

Try out [Case Management API v3](#) to interact with our new Case Management experience on Akamai Community.

Case Management API v2

Manage support requests to resolve issues with your Akamai applications and services.

Overview

The Case Management API helps you manage support requests to resolve any issues with your Akamai applications and services.

Who should use this API

Use this API if you are administer Akamai products and want to develop your own set of tools to interact with Akamai customer service representatives outside of [Akamai Control Center](#). Use this API to track the status of various support tickets, and to open new tickets. Use it to allow your own set of users to comment on support tickets and upload files that demonstrate problems. Use it also if you are a reseller who wants to develop a customer support layer for third-party users.

Get started

Before using the Beta API for the first time:

- Review [Get Started with APIs](#) for details on how to set up client tokens to access any Akamai API. These tokens appear as custom hostnames that look like this: `https://akzz-XXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXX.luna.akamaiapis.net`.
- To enable this API, choose the API service named Case Management, and set the access level to READ-WRITE.

After provisioning the API token, the user associated with the unique token can submit cases on behalf of other users.

Hypermedia

The API uses [links](#) to standard *hypermedia* objects to help clients locate related resources relevant within a set of JSON data. For example, the JSON below shows the sort of response you get when uploading a file attachment. Its [links](#) array may accommodate various related resources, but in this case presents only the case ticket under which the file is tracked. The [rel](#) represents the *link relation*, in this case a conceptual [parent](#) object, and the [href](#) that provides a navigable path to the resource once you prefix it with your hostname token.

```
{
  "caseId": "F-CS-1203600",
  "statusMessage": "Ticket has been successfully updated with file",
  "links": [
    {
      "href": "/case-management/v2/cases/F-CS-1203600",
      "rel": "parent"
    }
  ]
}
```

Resources

This section provides details on the Case Management API's various operations. The API supports two basic sets of operations, one workflow to create a new case, and another to add follow-up items to existing cases, including requests to close the case.

This workflow summarizes the operations necessary to create a new support ticket. See the steps for each operation for more detailed information.

1. First run [List Categories](#) to assign a top-level category.
2. Run [Get subcategories](#) to select one or more relevant subcategories for use in a new [POST object](#).
3. Run [Get case attributes](#) to prepare new data for the POST object.
4. Based on the [hypermedia](#) link embedded in response object, optionally run [Get a questionnaire](#) to answer questions relevant to the support ticket's top-level category.
5. Optionally run [Get severity definitions](#) and [Get response times](#) for help in assigning the appropriate severity level.
6. Once you have assembled all this information, run the [Create a New Case](#) operation.

Run the remaining [List cases](#) and [Get a case](#) operations to select support tickets. Run [Add a note](#) and

[Upload a file](#) to update a ticket, and [Download a File](#) to get uploaded files. Run [Close a Case](#) once the support ticket has been addressed.

API summary

Download the [RAML descriptors](#) for this API.

Operation	Method	Endpoint
Categories		
List Categories	GET	<code>/case-management/v2/categories</code>
Get subcategories	GET	<code>/case-management/v2/categories/{categoryType}/sub-categories</code>
Get case attributes	GET	<code>/case-management/v2/categories/{categoryType}/case-attributes{?problem,area,service,product}</code>
Get severity definitions	GET	<code>/case-management/v2/categories/{categoryType}/severity-definitions{?product}</code>
Get response times	GET	<code>/case-management/v2/categories/{categoryType}/response-time-frequency{?product}</code>
Questionnaires		
Get a questionnaire	GET	<code>/case-management/v2/questionnaires/{questionnaireId}</code>
Cases		
List cases	GET	<code>/case-management/v2/cases{?type,duration}</code>
Create a new case	POST	<code>/case-management/v2/cases</code>
Get a case	GET	<code>/case-management/v2/cases/{caseId}</code>
Add a note	POST	<code>/case-management/v2/cases/{caseId}/notes</code>
Close a case	POST	<code>/case-management/v2/cases/{caseId}/close-request</code>

[Upload a file](#) POST `/case-management/v2/cases/{caseId}/files`

[Download a file](#) GET `/case-management/v2/cases/{caseId}/files/{fileId}`

List categories

To create a new support ticket, you first need to categorize it, both with a top-level category and a set of subcategories. This operation gets the top-level category necessary to list the subcategories.

GET `/case-management/v2/categories`

Status 200 `application/json`

Object type: [Category](#)

Download schema: [Categories.json](#)

Response Body:

```
[
  {
    "categoryType": "Technical",
    "displayName": "Technical Support Issue",
    "description": "Performance, Stale Content, Errors and Unexpected behaviors, and other break/fix issues, etc."
  },
  {
    "categoryType": "Business",
    "displayName": "Business Support Issue or Question",
    "description": "CP Code, User Access and Management, How To Use Control Center, Reporting, Contract, etc."
  },
  {
    "categoryType": "Billing",
    "displayName": "Billing Support Issue or Question",
    "description": "Invoice issues, Billing discrepancies, Purchase Order details,
```

Run this operation to retrieve a `categoryType` value for use in gathering information to create a new support ticket:

1. GET `/case-management/v2/categories`.

2. From the response array, choose the appropriate [Category](#) object.
3. Store the object's `categoryType`.

Get subcategories

To create a new support ticket, you need to provide a top-level category and at least one relevant subcategory. This operation provides data from which to extract a subcategory type and unique value.

GET </case-management/v2/categories/{categoryType}/sub-categories>

Sample: </case-management/v2/categories/Technical/sub-categories>

Parameter	Type	Sample	Description
URL parameters			
<code>categoryType</code>	Enumeration	<code>Technical</code>	The top-level category assigned to the support ticket, either <code>Billing</code> , <code>Business</code> , <code>Professional-Services</code> , <code>Security</code> , or <code>Technical</code> .

Status 200 [application/json](#)

Object type: [SubCategory](#)

Download schema: [subCategory-200.json](#)

Response Body:

```
{
  "subCategory": {
    "subCategoryType": "area",
    "displayName": "Area",
    "subCategories": [
      {
        "subCategoryType": "CP Code Management",
        "displayName": "CP Code Management",
        "subCategory": {
          "subCategoryType": "service",
          "displayName": "Service",
          "subCategories": [
            {
```

```
        "subCategoryType": "Change Contract Association",
        "displayName": "Change Contract Association"
    },
    {
```

Run these steps to build an array of subcategories for use in creating a new support ticket.

1. Run the steps for the [List Categories](#) operation to store a relevant `categoryType` value.
2. Make a GET request to `/case-management/v2/categories/{categoryType}/sub-categories`.
3. The response object is a recursive data structure that represents a hierarchy of subcategories. Store the `subSelection` object's `subSelectionType` and `displayName`, which specify a topic for the subcategory.
4. If there is a `subCategories` array, it specifies each relevant subcategory within the topic. Select a relevant object from the array and store its `subSelectionType` as `subSelectionValue`.
5. If the subcategory features a nested `subSelection` member, optionally repeat the previous two steps until you have descended the tree to select the most appropriate pair of `subSelectionType` and `subSelectionValue` values.
6. Push an object with the `subSelectionType`, `subSelectionValue`, and `displayName` values to an array.
7. Optionally repeat if you want to assign more than one relevant subcategory to a support ticket.

Get case attributes

To create a new support ticket, this operation prompts for content you are expected to fill out. The response features various field names, expected data types, and enumerated values if appropriate.

GET `/case-management/v2/categories/{categoryType}/case-attributes{?problem,area,service,product}`

Sample: `/case-management/v2/categories/Technical/case-attributes?problem=Alerts&area=Company%20Registration&service=Address%20Change&product=Alta`

Parameter	Type	Sample	Description
URL parameters			

categoryType Enumeration **Technical** The top-level category assigned to the support ticket, either **Billing**, **Business**, **Professional-Services**, **Security**, or **Technical**.

Optional query parameters

area String **Company Registration** Filter results to subcategories whose **subCategoryType** is **area**, and where the parameter value matches the **subCategoryValue**.

problem String **Alerts** Filter results to subcategories whose **subCategoryType** is **problem**, and where the parameter value matches the **subCategoryValue**.

product String **Alta** Filter results to subcategories whose **subCategoryType** is **product**, and where the parameter value matches the **subCategoryValue**.

service String **Address Change** Filter results to subcategories whose **subCategoryType** is **service**, and where the parameter value matches the **subCategoryValue**.

Status 200 **application/json**

Object type: [CaseAttributes](#)

Download schema: [CaseAttributes.json](#)

Response Body:

```
{
  "userDetail": {
    "userName": {
      "value": "Sample User",
      "uiControls": {
        "fieldType": "text",
        "displayName": "Name",
        "maxLength": 100,

```

```
        "isRequired": true,
        "isDisabled": false
    },
    },
    "userPhone": {
        "value": 12345467890,
        "uiControls": {
            "fieldType": "text",
            "displayName": "Phone Number",
```

Run these steps to get relevant values to add when [POSTing a new support ticket](#).

1. Run the steps for the [List Categories](#) operation to store a relevant `categoryType` value.
2. Optionally set the `area`, `problem`, `product` or `service` parameters to filter attributes to a relevant type of subcategory.
3. Make a GET request to `/case-management/v2/categories/{categoryType}/case-attributes{?area,problem,product,service}`.
4. The response object specifies various field names, such as `subject` and `description`, to include in the POST object when creating a new support ticket. Iterate over the keys and corresponding object for these fields and other fields in the nested `userDetail` object.
5. For each object, optionally present the `value` as an interface prompt.
6. For each object, optionally check the `uiControls.fieldType` for guidance on the expected data type.
7. For each object, if an `answerOptions` array is present, it collects fixed or previously cached values. Optionally choose the appropriate `answerValue` from one of the objects.
8. For the `severity` object, optionally navigate its `links` for more information on severity levels. Make a GET request on these to run the [Get severity definitions](#) and [Get Response Times](#) operations.
9. To include optional questionnaire items in the new support ticket, make a GET request on the `href` for the top-level hypermedia `links`. Add a `questionnaire` member to the POST object based on the steps for the [Get a questionnaire](#) operation.

Once you have iterated through the fields for each case attribute, your [POST](#) object should be filled with appropriate top-level values, and a `userDetail` object should capture information about the user to whom the support ticket applies. An optional `questionnaire` should capture relevant

questions and their answers.

Get severity definitions

To provide guidance when creating a new ticket, this operation provides details on each severity level.

GET `/case-management/v2/categories/{categoryType}/severity-definitions{?product}`

Sample: `/case-management/v2/categories/Technical/severity-definitions?product=Alta`

Parameter	Type	Sample	Description
URL parameters			
<code>categoryType</code>	Enumeration	<code>Technical</code>	The top-level category assigned to the support ticket, either <code>Billing</code> , <code>Business</code> , <code>Professional-Services</code> , <code>Security</code> , or <code>Technical</code> .
Optional query parameters			
<code>product</code>	String	<code>Alta</code>	Filter results to subcategories whose <code>subCategoryType</code> is <code>product</code> , and where the parameter value matches the <code>subCategoryValue</code> .

Status 200 `application/json`

Object type: [SeverityDefinition](#)

Download schema: [SeverityDefinition.json](#)

Response Body:

```
[
  {
    "severityLevelType": "1-Critical",
    "description": "An existing network or environment is down or there is a critical impact to an End User's business operation. There is no reasonable workaround currently available.",
    "escalationInstructions": "Open a ticket online in Control Center and optionally call +1-855-625-2624 (1-855-6-AKAMAI) or +1-617-444-4605",
```

```

    "standardResponseTime": "1 hour"
  },
  {
    "severityLevelType": "2-Major",
    "description": "Operation of customer's existing network or environment is severely degraded or significant aspects of End User's business operation are negatively impacted by unacceptable network or environment performance.",
    "escalationInstructions": "Open a ticket online in Control Center and optionally call +1-855-625-2624 (1-855-6-AKAMAI) or +1-617-444-4605",
  }
}

```

Run these steps when creating a new support ticket only to get information about the possible **severity** level.

1. Run the steps for the [List Categories](#) operation to store a relevant **categoryType** value.
2. Optionally set **product** to filter results relevant only to a specific product.
3. Make a GET request to `/case-management/v2/categories/{categoryType}/severity-definitions{?product}`.
4. From the response array, select the object whose **severityLevelType** matches the proposed **severity** you may assign. The **description** characterizes the type of problem, while the **escalationInstructions** suggests alternate ways to contact Akamai about the problem.

The response's **standardResponseTime** offers an estimate for how long it takes to address the ticket, but you can run the [Get Response Times](#) operation for a more detailed estimate.

Get response times

To provide guidance when creating a new ticket, this operation provides estimates for response times for each severity level.

GET `/case-management/v2/categories/{categoryType}/response-time-frequency{?product}`

Sample: `/case-management/v2/categories/Technical/response-time-frequency?product=Alta`

Parameter	Type	Sample	Description
URL parameters			
categoryType	Enumeration	Technical	The top-level category assigned to the support ticket, either Billing , Business , Professional-Services , Security , or

Technical.

Optional query parameters

<code>product</code>	String	<code>Alta</code>	Filter results to subcategories whose <code>subCategoryType</code> is <code>product</code> , and where the parameter value matches the <code>subCategoryValue</code> .
----------------------	--------	-------------------	--

Status 200 `application/json`

Object type: [ResponseTime](#)

Download schema: [ResponseTime.json](#)

Response Body:

```
{
  "additionalInformation": [
    "Initial Response = First communication from Akamai personnel to customer by phone or email",
    "Status Updates = Any subsequent Akamai communication to customer by phone or email",
    "* This 1-Major Impact SLA is applicable only when the issue is opened via phone"
  ],
  "responseTimeAndFrequencyDetails": [
    {
      "supportTypeDisplayName": "Premium & Premium 2.0",
      "interactionTypes": [
        {
          "interactionType": "initialResponse",
          "responseTimes": [
            {
```

Run these steps when creating a new support ticket only to get information about the possible `severity` level.

1. Run the steps for the [List Categories](#) operation to store a relevant `categoryType` value.
2. Optionally set `product` to filter results relevant only to a specific product.
3. Make a GET request to `/case-management/v2/categories/{categoryType}/response-time-frequency{?product}`

4. From the response's `responseTimeAndFrequencyDetails` array, select the appropriate type of support object based on the `supportTypeDisplayName`.
5. From the support object's `interactionTypes` array, choose the interaction with the most appropriate `interactionType`, either `initialResponse` or `statusUpdates`.
6. From the `responseTimes` array, choose the most appropriate `severityLevelType` to consider when posting a `severity` level for a new ticket. The `responseTime` provides an estimated time, expressed as plain text.

Get a questionnaire

Provides a series of questions and answers relevant to the case category, for use in creating a new ticket.

GET `/case-management/v2/questionnaires/{questionnaireId}`

Sample: `/case-management/v2/questionnaires/100`

Parameter	Type	Sample	Description
URL parameters			
<code>questionnaireId</code>	String	<code>100</code>	Unique identifier for a questionnaire.

Status 200 `application/json`

Object type: [Questionnaire](#)

Download schema: [Questionnaire.json](#)

Response Body:

```
{
  "questionnaireId": 100,
  "questions": [
    {
      "questionId": 670,
      "questionText": "Question for Id 670",
      "helpText": "Help text for 670",
```

```
"currentAnswers": [  
  "answer for question 670"  
],  
"uiControls": {  
  "fieldType": "Date",  
  "isRequired": true,  
  "isDisabled": false,  
  "isClearable": true  
}  
},
```

These steps produce a `questionnaire` object suitable to include within the POST request for a new support ticket. You run this operation based on a [hypermedia](#) link embedded within the response object for the [Get Case Attributes](#) operation. The required `questionnaireId` necessary to access the `/case-management/v2/questionnaires/{questionnaireId}` resource is available only within this category-specific context.

1. Make a GET request on the supplied `href` link.
2. Store the response's `questionnaireId` in a `questionnaire` object.
3. For each item in the `questions` array, evaluate the `questionText` and `helpText`. If the question is relevant, store the `questionId`, otherwise skip to the next question.
4. Check the `uiControls.fieldType` for the answer's expected data type.
5. Iterate over any available `answerOptions` array within the question, and optionally push its `answerValue` to a `currentAnswers` array. Otherwise push another value based on the data type.
6. Store the `questionId` string and the `currentAnswers` array in an object, and push the object to the `questionnaire` object's `questions` array.
7. Within an `answerOptions` item, optionally answer any series of nested `dependentQuestions` as described above. These questions are relevant only to the `answerOptions` item.

After running these steps, the `questionnaire` object should be ready to include in a larger POST object that creates a new support ticket.

List cases

This operation lists support tickets and allows you to filter the results.

GET `/case-management/v2/cases{?type,duration}`

Sample: `/case-management/v2/cases?type=user-active&duration=180`

Parameter	Type	Sample	Description
Optional query parameters			
<code>duration</code>	String	<code>180</code>	Returns tickets created within the specified number of days that match a state defined in the <code>type</code> parameter. For requests with the <code>type</code> parameter ignored or set to <code>all</code> , it returns tickets created within the specified number of days that match any of the states in the <code>type</code> parameter: <code>user-active</code> , <code>user-closed</code> , <code>company-active</code> , or <code>company-closed</code> .
<code>type</code>	Enumeration	<code>user-active</code>	Filter tickets, either <code>user-active</code> or <code>user-closed</code> for the user filing the ticket, <code>company-active</code> or <code>company-closed</code> for cases associated with the API token's account, or unfiltered <code>all</code> by default.

Status 200 `application/json`

Object type: [Case](#)

Download schema: [Cases.json](#)

Response Body:

```
[
  {
    "caseId": "F-CS-1013252",
    "subject": "[Managedsecurity] Customers Westpac and NAB cannot view open cases - support service not included in the contract under Control Center",
    "categoryType": "Business",
    "status": "Unassigned",
    "createdDate": "2016-02-29T14:15:40+0000",
    "modifiedDate": "2016-02-29T14:15:40+0000",
    "assignedTo": "In Queue",
    "customer": "netflex",
    "submittedBy": "Employee xyz",
    "severity": "3-Low Impact",
```

```
"links": [  
  {  
    "rel": "self",  
    "href": "/case-management/v2/cases/F-CS-1013252"  }  
]
```

1. Optionally choose a `type` of case on which to filter: `company-active` or `company-closed` for account-wide tickets, otherwise `user-active` or `user-closed` for tickets associated with the API token.
2. Make a GET request to `/case-management/v2/cases/{?type}`.

From the resulting response array, choose a [Case.ticketInfo](#) object whose `subject` matches the appropriate support ticket. To access details about the case, store its `caseId` and run the [Get a case](#) operation, or else navigate directly to the [hypermedia link](#).

Create a new case

Create a new support ticket based on a chosen category and set of subcategories.

POST `/case-management/v2/cases`

Content-Type: `application/json`

Object type: [CasePost](#)

Download schema: [CasePost.json](#)

Request Body:

```
{  
  "severity": "1-Major Impact",  
  "subject": "Subject",  
  "description": "Test Description",  
  "customer": "telefilm",  
  "customerTrackingNumber": "TRACK-FSDJW452",  
  "endCustomerName": "Customer 1",  
  "partnerTicketNumber": "SDG232ASD23",  
  "categoryType": "Technical",  
  "doNotShowInPortalFlag": false,  
  "questionnaire": {  
    "questionnaireId": "100",  
    "questions": [  
      {  
        "questionId": "670",
```

```
"currentAnswers": [  
  "2015-11-29T11:58:53.273Z"
```

Status 200 `application/json`

Object type: [Response](#)

Download schema: `created-case-200.json`

Response Body:

```
{  
  "success": true,  
  "caseId": "F-CS-1778108",  
  "message": "Case has been successfully created with Case Id F-CS-1778108",  
  "details": [  
    {  
      "type": "diagnostic-url",  
      "url": "http://www.ipdiagnostics.com/request.action?client=akamai&case=F-CS-1778108&URL=https://www.foo.com/d "  
    }  
  ]  
}
```

1. Run the [List Categories](#) operation and store the relevant `categoryType`.
2. Create a [POST object](#), assigning the `categoryType` to the top level.
3. Run the [Get subcategories](#) operation and follow the operations to build an array of objects featuring `subCategoryType`, `subCategoryValue`, and `displayName` members. Place this array in the POST object's `subCategories`.
4. Run [Get case attributes](#) to prepare new data for the POST object. In addition to top-level data, it should also feature data about the user within a `userDetail` object.
5. Based on the [hypermedia](#) link embedded in the [CaseAttributes](#) response object, optionally run [Get a questionnaire](#) to answer questions relevant to the support ticket's top-level category.
6. Optionally run [Get severity definitions](#) for help in clarifying severity levels.
7. Optionally run [Get response times](#) for help in estimating how long a response will take.
8. POST the object to `/case-management/v2/cases/`.

- From the successful [response object](#), optionally store the `caseId` that identifies the new support ticket.

Get a case

This retrieves the complete details about a specific case.

GET `/case-management/v2/cases/{caseId}`

Sample: `/case-management/v2/cases/F-CS-1013252`

Parameter	Type	Sample	Description
URL parameters			
<code>caseId</code>	String	<code>F-CS-1013252</code>	Unique identifier for the support ticket.

Status 200 `application/json`

Object type: [Case](#)

Download schema: [Case.json](#)

Response Body:

```
{
  "ticketInfo": {
    "caseId": "F-CS-1455428",
    "subject": "testing",
    "description": "testing",
    "status": "Unassigned",
    "severity": "3-Low Impact",
    "assignedTo": "In Queue",
    "categoryType": "Technical",
    "createdDate": "2016-02-29T14:15:40+0000",
    "modifiedDate": "2016-03-09T08:57:13+0000",
    "customer": "netflix",
    "externalSystemNumber": "AD-1432",
    "geoLinkId": "1-HGAJ",
    "partnerTicketNumber": "ADYH-1272",
    "endCustomerName": "a-netflix",
    "nextStepExternal": "NA",
```

1. Run the [List cases](#) operation and select the appropriate object from the response array. Either navigate directly to the embedded [hypermedia link](#), or store the `caseId`.
2. If storing the `caseId` value, make a GET request to `/case-management/v2/cases/{caseId}`.

Add a note

Once a support ticket has been created, this operation adds follow-up comments, which are reflected in the Case object's `activities` array.

POST `/case-management/v2/cases/{caseId}/notes`

Sample: `/case-management/v2/cases/F-CS-1013252/notes`

Content-Type: `application/json`

Download schema: [add-note-to-a-case.json](#)

Request Body:

```
{
  "comment": "Adding note"
}
```

Parameter	Type	Sample	Description
URL parameters			
<code>caseId</code>	String	<code>F-CS-1013252</code>	Unique identifier for the support ticket.

Status 200 `application/json`

Object type: [Response](#)

Download schema: [added-note-to-a-case-200.json](#)

Response Body:

```
{
  "caseId": "F-CS-1203600",
  "statusMessage": "Ticket has been successfully updated with notes",
  "links": [
```

```
{
  {
    "href": "/case-management/v2/cases/F-CS-1203600",
    "rel": "parent"
  }
}
```

1. If you do not already have a relevant `caseId` value, run the [List cases](#) operation, select the appropriate object from the response array, and store the `caseId`.
2. Create a simple request object featuring a single `comment` member and accompanying text. For example: `{"comment":"This is the comment."}`
3. POST the object to `/case-management/v2/cases/{caseId}/notes`.

Close a case

This operation requests to close a ticket, without directly closing it.

POST `/case-management/v2/cases/{caseId}/close-request`

Sample: `/case-management/v2/cases/F-CS-1013252/close-request`

Content-Type: `application/json`

Download schema: [request-close-to-a-case.json](#)

Request Body:

```
{
  "comment": "Issue has been resolved.Kindly close the ticket"
}
```

Parameter	Type	Sample	Description
URL parameters			
<code>caseId</code>	String	<code>F-CS-1013252</code>	Unique identifier for the support ticket.

Status 200 `application/json`

Object type: [Response](#)

Download schema: [requested-close-to-a-case-200.json](#)

Response Body:

```
{
  "caseId": "F-CS-1203600",
  "statusMessage": "Ticket has been successfully updated with notes",
  "links": [
    {
      "href": "/case-management/v2/cases/F-CS-1203600",
      "rel": "parent"
    }
  ]
}
```

1. If you do not already have a relevant `caseId` value, run the [List Cases](#) operation, select the appropriate object from the response array, and store the `caseId`.
2. Create a simple request object featuring a single `comment` member and accompanying text. For example: `{"comment":"please close the ticket."}`
3. POST the comment object to `/case-management/v2/cases/{caseId}/close-request`.

Upload a file

Once a support ticket has been created, this operation adds a file attachment and comment to it, which are reflected in the Case object's `activities` array.

POST `/case-management/v2/cases/{caseId}/files`

Sample: `/case-management/v2/cases/F-CS-1013252/files`

Content-Type: `multipart/form-data`

See Parameters for details on upload content.

Parameter	Type	Sample	Description
URL parameters			

<code>caseId</code>	String	<code>F-CS-1013252</code>	Unique identifier for the support ticket.
---------------------	--------	---------------------------	---

Form parameters

<code>file</code>	File	<code>screen_shot.png</code>	The file to be uploaded
-------------------	------	------------------------------	-------------------------

<code>comment</code>	String	<code>Adding logs</code>	Any additional information
----------------------	--------	--------------------------	----------------------------

Status 200 `application/json`

Object type: [Response](#)

Download schema: [uploaded-file-to-a-case-200.json](#)

Response Body:

```
{
  "caseId": "F-CS-1203600",
  "statusMessage": "Ticket has been successfully updated with notes",
  "links": [
    {
      "href": "/case-management/v2/cases/F-CS-1203600",
      "rel": "parent"
    }
  ]
}
```

1. If you do not already have a relevant `caseId` value, run the [List Cases](#) operation, select the appropriate object from the response array, and store the `caseId`.
2. Create a simple request object featuring a single `comment` member and accompanying text. For example: `{"comment": "Description of file."}`
3. Encode the file if it is in a binary format.
4. Form a `multipart/form-data` string, using a unique separator to delimit each part. The data needs to incorporate the comment object typed as `application/json`, and the file upload string using its appropriate MIME type.
5. POST the `multipart/form-data` typed string to `/case-management/v2/cases/{caseId}/files`.

- From the successful [response object](#), optionally store the [hypermedia link](#) to download the file later.

Download a file

This operation downloads any file attached to a support ticket.

GET `/case-management/v2/cases/{caseId}/files/{fileId}`

Sample: `/case-management/v2/cases/F-CS-1013252/files/00Pm0000009XDiiEAG`

Parameter	Type	Sample	Description
URL parameters			
<code>caseId</code>	String	<code>F-CS-1013252</code>	Unique identifier for the support ticket.
<code>fileId</code>	String	<code>00Pm0000009XDiiEAG</code>	Unique identifier for the file attachment.

Status 200 `application/octet-stream`

Response Body:

```
4944 3302 0000 0000 1030 3230 3134 0000
0826 0000 1000 656e 6769 5475 6e50 4741
5000 3000 0054 454e 0000 0f00 6954 756e
6573 2076 372e 342e 3200 434f 4d00 0068
0065 6e67 6954 756e 4e4f 524d 0020 3030
3030 3032 3735 2030 3030 3030 3239 4320
3030 3030 3433 3334 2030 3030 3034 3138
3020 3030 3032 3539 4131 2030 3030 3235
3941 3120 3030 3030 3737 4242 2030 3030
3036 4243 3520 3030 3031 4538 3738 2030
3030 3145 3830 4600 434f 4d00 0082 0065
6e67 6954 756e 534d 5042 0020 3030 3030
3030 3030 2030 3030 3030 3231 3020 3030
3030 3039 3446 2030 3030 3030 3030 3030
3037 3237 4132 3120 3030 3030 3030 3030
2030 3033 4533 4638 3320 3030 3030 3030
3030 2030 3030 3030 3030 3020 3030 3030
```

Ideally, when running the [Upload a file](#) operation, you should store the resulting [hypermedia link](#) to later access the file. Otherwise follow these steps if you don't have that information.

1. If you do not already have a relevant `caseId` value, run the [List cases](#) operation, select the appropriate object from the response array, and store the `caseId`.
2. Run the [Get a case](#) operation by making a GET request to `/case-management/v2/cases/{caseId}`.
3. Filter the response [Case](#) object's `activities` array to include objects whose `activityType` is `attachment`.
4. Select the activity whose `fileName` or `activityCreatedDate` is most appropriate.
5. Make a GET request on the embedded [hypermedia link](#). This corresponds to this URL template, for which there is no direct need to store the `fileId` value: `/case-management/v2/cases/{caseId}/files/{fileId}`.

Data

This section provides details for each type of data object the API exchanges. The [Case](#) object specifies the response when you GET an individual support ticket. The [CasePost](#) object specifies the form of request necessary when creating new support tickets. The [UiControls](#) is a common object that appears embedded within other object types. The [Response](#) is a common object that acknowledges comments added to a support ticket, file uploads, or requests to close the ticket.

Download the [JSON schemas](#) for this API.

The data schema tables below list membership requirements as follows:

-
- ✓ Member is required in requests, or always present in responses, even if its value is empty or `null`.
 - Member is optional, and may be omitted in some cases.
-

Category

For use in creating new support tickets, this specifies high-level categories that assign questionnaires and other inputs.

Download schema: [Category.json](#)

Sample GET:

```
[
  {
    "categoryType": "Technical",
    "displayName": "Technical Support Issue",
    "description": "Performance, Stale Content, Errors and Unexpected behaviors, and other break/fix issues, etc."
  },
  {
    "categoryType": "Business",
    "displayName": "Business Support Issue or Question",
    "description": "CP Code, User Access and Management, How To Use Control Center, Reporting, Contract, etc."
  },
  {
    "categoryType": "Billing",
    "displayName": "Billing Support Issue or Question",
    "description": "Invoice issues, Billing discrepancies, Purchase Order details,
```

Category members

Member	Type	Required	Description
Category: For use in creating new support tickets, this specifies high-level categories that assign questionnaires and other inputs.			
<code>categoryType</code>	Enumeration	✓	The category type, either Billing , Business , Professional-Services , Security , or Technical .
<code>description</code>	String	○	Provides additional descriptive text for the category.
<code>displayName</code>	String	✓	The name of the category, for use in display within the interface.

SubCategory

Encapsulates a hierarchy of subcategories based on the initially selected [Category](#). This recursive data structure, accessible through the response's top-level `subCategory` member, represents two levels of contextual data. One provides a common subcategory *topic* for use in interfaces, and another provides a *value* for each subcategory to assign when creating a new support ticket. Objects that represent these topics always feature a nested `subCategories` array. Objects that represent values within that array may feature an additional `subCategory` encapsulating the same nested topic/value structure. If neither `subCategories` nor `subCategory` members are present, the object is a subcategory value that serves as a terminal leaf node with no nested subcategories.

Download schema: [SubCategory.json](#)

Sample GET:

```
{
  "subCategory": {
    "subCategoryType": "area",
    "displayName": "Area",
    "subCategories": [
      {
        "subCategoryType": "CP Code Management",
        "displayName": "CP Code Management",
        "subCategory": {
          "subCategoryType": "service",
          "displayName": "Service",
          "subCategories": [
            {
              "subCategoryType": "Change Contract Association",
              "displayName": "Change Contract Association"
            },
            {

```

SubCategory members

Member	Type	Required	Description
SubCategory			Encapsulates a hierarchy of subcategories based on the initially selected Category . This recursive data structure, accessible through the response's top-level subCategory member, represents two levels of contextual data. One provides a common subcategory <i>topic</i> for use in interfaces, and another provides a <i>value</i> for each subcategory to assign when creating a new support ticket. Objects that represent these topics always feature a nested subCategories array. Objects that represent values within that array may feature an additional subCategory encapsulating the same nested topic/value structure. If neither subCategories nor subCategory members are present, the object is a subcategory value that serves as a terminal leaf node with no nested subcategories.
displayName	String	✓	Interface display text that corresponds to the subCategoryType .
subCategories	Sub Category array	○	Specifies a series of nested SubCategory objects that represent values. Required within objects that serve as the subcategory's topic.
subCategory	Object	○	Within a subcategory value, specifies an optional nested subcategory topic with its own set of

nested values.

subCategoryType	String	✓	The subcategory topic, or the value to assign to a new support ticket.
---------------------------------	--------	---	--

CaseAttributes

Specifies various details to fill out when creating a new support ticket, along with additional interface tools to help validate the input. Encapsulates information about the user who creates the ticket, and possibly other customers and partners for whom they are creating ticket.

Download schema: [CaseAttributes.json](#)

Sample GET:

```
{
  "userDetail": {
    "userName": {
      "value": "Sample User",
      "uiControls": {
        "fieldType": "text",
        "displayName": "Name",
        "maxlength": 100,
        "isRequired": true,
        "isDisabled": false
      }
    },
    "userPhone": {
      "value": 12345467890,
      "uiControls": {
        "fieldType": "text",
        "displayName": "Phone Number",
```

CaseAttributes members

Member	Type	Required	Description
CaseAttributes			Specifies various details to fill out when creating a new support ticket, along with additional interface tools to help validate the input. Encapsulates information about the user who creates the ticket, and possibly other customers and partners for whom they are creating ticket.
additionalDetails	Case Attributes.additionalDetails	○	Provides additional text that describes the problem.

customer	Case Attributes.customer	○	Encapsulates information about the customer on whose behalf the user is entering the support ticket.
customer TrackingNumber	Case Attributes.customer Tracking Number	○	Provides a key for a customer's separately provisioned support ticket.
description	Case Attributes.description	○	Provides details for the issue to resolve.
endCustomerName	Case Attributes.end Customer Name	○	Inputs the name of the customer on whose behalf the user is entering a support ticket.
links	Hypermedia array	○	Provides a location for this ticket's questionnaire.
partnerTicket Number	Case Attributes.partner Ticket Number	○	Provides a key for a partner's separately provisioned support ticket.
severity	Case Attributes.severity	✓	Assigns a severity level to the support ticket.
subject	Case Attributes.subject	✓	Provides a title for the support ticket.
upload	Case Attributes.upload	○	Provides the option to upload a file within a user interface. It does not function within the API, and always appears as null .
userDetail	Case Attributes.userDetail	✓	Encapsulates information about the user opening the support ticket.

CaseAttributes.additionalDetails: Provides additional text that describes the problem.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	✓	Provides additional text that describes the problem.
--------------	--------	---	--

CaseAttributes.customer: Encapsulates information about the customer on whose behalf the user is entering the support ticket.

answerOptions	Case Attributes.customer.answer Options[]	✓	Provides options based on previously supplied customer information.
----------------------	---	---	---

currentAnswers	Array	✓	When creating a support ticket, this provides the answers to the Questionnaire , which is accessed separately from the main set of case attributes. Values correspond to answerId strings.
-----------------------	-------	---	---

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

CaseAttributes.customer.answerOptions []: Provides options based on previously supplied customer information.

answerId	String	✓	Identifies each customer.
-----------------	--------	---	---------------------------

answerValue	String	✓	Displays the customer's name.
--------------------	--------	---	-------------------------------

CaseAttributes.customerTrackingNumber: Provides a key for a customer's separately provisioned support ticket.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	✓	Provides a key for a customer's separately provisioned support ticket.
--------------	--------	---	--

CaseAttributes.description: Provides details for the issue to resolve.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	✓	Provides details for the issue to resolve.
--------------	--------	---	--

CaseAttributes.endCustomerName: Inputs the name of the customer on whose behalf the user is entering a support ticket.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	✓	Inputs the name of the customer on whose behalf the user is entering a support ticket.
--------------	--------	---	--

CaseAttributes.partnerTicketNumber: Provides a key for a partner's separately provisioned support ticket.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	✓	Provides a key for a partner's separately provisioned support ticket.
--------------	--------	---	---

CaseAttributes.severity: Assigns a severity level to the support ticket.

answerOptions	Case Attributes.severity.answer Options[]	✓	Provides a series of options from which to choose a severity level.
----------------------	---	---	---

links	Hypermedia array	✓	Locates related resources that describe the severity level and provide estimated response times.
--------------	----------------------------------	---	--

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

value	String	○	Selects the severity level.
--------------	--------	---	-----------------------------

CaseAttributes.severity.answerOptions []: Provides a series of options from which to choose a severity level.

answerId	String	✓	Identifies the severity option.
-----------------	--------	---	---------------------------------

answerValue	String	✓	Provides display text for the severity option.
--------------------	--------	---	--

CaseAttributes.subject: Provides a title for the support ticket.

uiControls	UiControls	✓	Specifies a set of controls to input the value .
-------------------	----------------------------	---	---

		✓	
--	--	---	--

value String Provides a title for the support ticket.

CaseAttributes.upload: Provides the option to upload a file within a user interface. It does not function within the API, and always appears as **null**.

uiControls [UiControls](#) ✓ Specifies a set of controls to input the **value**.

value Null ✓ Provides the option to upload a file. The API value appears as **null**.

CaseAttributes.userDetail: Encapsulates information about the user opening the support ticket.

userCompany [Case Attributes.userDetail.user Company](#) ○ Encapsulates information about the user's company.

userEmail [Case Attributes.userDetail.userEmail](#) ○ Encapsulates the user's email address.

userName [Case Attributes.userDetail.userName](#) ○ Encapsulates the name of the user.

userPhone [Case Attributes.userDetail.userPhone](#) ○ Encapsulates the user's phone number.

CaseAttributes.userDetail.userCompany: Encapsulates information about the user's company.

uiControls [UiControls](#) ✓ Specifies a set of controls to input the **value**.

value String ✓ Inputs the user's company name.

CaseAttributes.userDetail.userEmail: Encapsulates the user's email address.

uiControls [UiControls](#) ✓ Specifies a set of controls to input the **value**.

<code>value</code>	String	✓	Inputs the email address.
<code>CaseAttributes.userDetail.userName</code> : Encapsulates the name of the user.			
<code>uiControls</code>	UiControls	✓	Specifies a set of controls to input the <code>value</code> .
<code>value</code>	String	✓	Inputs the name of the user.
<code>CaseAttributes.userDetail.userPhone</code> : Encapsulates the user's phone number.			
<code>uiControls</code>	UiControls	✓	Specifies a set of controls to input the <code>value</code> .
<code>value</code>	Integer	✓	Inputs the phone number, expressed as an integer.

ResponseTime

For a given Support Type and Severity level type, specifies initial response times and frequency of updates for the created case.

Download schema: [ResponseTime.json](#)

Sample GET:

```
{
  "additionalInformation": [
    "Initial Response = First communication from Akamai personnel to customer by phone or email",
    "Status Updates = Any subsequent Akamai communication to customer by phone or email",
    "* This 1-Major Impact SLA is applicable only when the issue is opened via phone"
  ],
  "responseTimeAndFrequencyDetails": [
    {
      "supportTypeDisplayName": "Premium & Premium 2.0",
      "interactionTypes": [
        {
          "interactionType": "initialResponse",
          "responseTimes": [
            {
```

ResponseTime members

Member	Type	Required	Description
--------	------	----------	-------------

ResponseTime: For a given Support Type and Severity level type, specifies initial response times and frequency of updates for the created case.

additionalInformation	Array	✓	Text items relevant only within an interface.
------------------------------	-------	---	---

responseTimeAndFrequencyDetails	ResponseTime.responseTimeAndFrequencyDetails[]	✓	Associates support levels, severity levels, and response times.
--	--	---	---

ResponseTime.responseTimeAndFrequencyDetails []: Associates support levels, severity levels, and response times.

interactionTypes	ResponseTime.responseTimeAndFrequencyDetails[].interactionTypes[]	✓	For a given support level, lists interaction categories.
-------------------------	---	---	--

supportTypeDisplayName	String	✓	Display text corresponding to a support-level category.
-------------------------------	--------	---	---

ResponseTime.responseTimeAndFrequencyDetails [].interactionTypes []: For a given support level, lists interaction categories.

interactionType	Enumeration	✓	The type of support interaction, either initialResponse or statusUpdates .
------------------------	-------------	---	--

responseTimes	ResponseTime.responseTimeAndFrequencyDetails[].interactionTypes[].responseTimes[]	✓	Associates support levels with estimated response times.
----------------------	---	---	--

ResponseTime.responseTimeAndFrequencyDetails [].interactionTypes [].responseTimes []: Associates support levels with estimated response times.

responseTime	String	✓	Estimated response time, represented as plain text.
---------------------	--------	---	---

severityLevelType	String	✓	Specifies the level of severity, corresponding
--------------------------	--------	---	--

to the support ticket's [severity](#).

SeverityDefinition

For use in helping the user assign a severity level, this object encapsulates details about each.

Download schema: [SeverityDefinition.json](#)

Sample GET:

```
[
  {
    "severityLevelType": "1-Critical",
    "description": "An existing network or environment is down or there is a critical impact to an End User's business operation. There is no reasonable workaround currently available.",
    "escalationInstructions": "Open a ticket online in Control Center and optionally call +1-855-625-2624 (1-855-6-AKAMAI) or +1-617-444-4605",
    "standardResponseTime": "1 hour"
  },
  {
    "severityLevelType": "2-Major",
    "description": "Operation of customer's existing network or environment is severely degraded or significant aspects of End User's business operation are negatively impacted by unacceptable network or environment performance.",
    "escalationInstructions": "Open a ticket online in Control Center and optionally call +1-855-625-2624 (1-855-6-AKAMAI) or +1-617-444-4605",
```

SeverityDefinition members

Member	Type	Required	Description
SeverityDefinition			For use in helping the user assign a severity level, this object encapsulates details about each.
description	String	✓	Describes the types of problem ordinarily associated with the level of severity.
escalationInstructions	String	✓	For use in interfaces, suggests other ways to get help.
severityLevelType	String	✓	Provides the key that identifies the severity level.

standard
ResponseTime

String



Provides an estimate for the amount of time it typically takes to respond to the support request, expressed as a string such as **1 business day**.

Questionnaire

Encapsulates questionnaire data to help categorize problems when opening support tickets.

Download schema: [Questionnaire.json](#)

Sample GET:

```
{
  "questionnaireId": 100,
  "questions": [
    {
      "questionId": 670,
      "questionText": "Question for Id 670",
      "helpText": "Help text for 670",
      "currentAnswers": [
        "answer for question 670"
      ],
      "uiControls": {
        "fieldType": "Date",
        "isRequired": true,
        "isDisabled": false,
        "isClearable": true
      }
    }
  ],
},
```

Questionnaire members

Member	Type	Required	Description
Questionnaire			Encapsulates questionnaire data to help categorize problems when opening support tickets.
questionnaireId	Integer	✓	Unique identifier for the questionnaire.
questions	Questionnaire.questions[]	✓	Encapsulates each question presented to the user.
Questionnaire.questions[]			Encapsulates each question presented to the user.
answerOptions	Questionnaire.	○	Specifies a set of multiple choice answers.

[questions\[\]](#).
[answer](#)
[Options\[\]](#)

currentAnswers	Array	○	Specifies the selected answer.
helpText	String	○	Provides any additional context for the question.
questionId	Integer	✓	Unique integer identifier for each question.
questionText	String	✓	The question's display text.
uiControls	UiControls	✓	Specifies a set of controls to input the value .
Questionnaire.questions [].answerOptions [] : Specifies a set of multiple choice answers.			
answerId	Integer	✓	Unique integer identifier for each answer.
answerValue	String	✓	The answer's display text.
dependent Questions	Object	○	Specifies a set of already-answered questions whose currentAnswers would determine whether it's appropriate to present this answer as an option.

UiControls

This object, used to present [Questionnaire](#) and [CaseAttributes](#) data within a user interface, specifies form elements to gather user input.

Download schema: [UiControls.json](#)

UiControls members

Member	Type	Required	Description
UiControls : This object, used to present Questionnaire and CaseAttributes data within a user interface, specifies form elements to gather user input.			
displayName	String	○	The interface label that displays alongside the form input.

<code>fieldType</code>	Enumeration	✓	Specifies the interface field's semantic data type, either <code>checkbox</code> , <code>date-time</code> , <code>date-time-timezone</code> , <code>decimal</code> , <code>dropdown</code> , <code>file-upload</code> , <code>number</code> , <code>radio</code> , <code>text</code> , or <code>textarea</code> .
<code>isClearable</code>	Boolean	○	Whether the value of the input element can be modified. Applies only when the <code>fieldType</code> is <code>date-time</code> , <code>date-time-timezone</code> , or <code>dropdown</code> .
<code>isDisabled</code>	Boolean	✓	Whether the input element is available within the interface.
<code>isRequired</code>	Boolean	✓	Whether the data must be provided.
<code>maxLength</code>	Integer	○	When the <code>fieldType</code> is <code>text</code> or <code>textarea</code> , this specifies the maximum number of characters allowed.
<code>maxSize</code>	String	○	When the <code>fieldType</code> is <code>file-upload</code> , this specifies the maximum size of the file, formatted to specify units such as <code>5MB</code> .
<code>rows</code>	Integer	○	When the <code>fieldType</code> is <code>textarea</code> , this specifies the number of vertical lines that display.

CasePost

Encapsulates data necessary to create a new support ticket. Once created, much of this data is reflected back in the `ticketInfo` object.

Download schema: [CasePost.json](#)

Sample GET:

```
{
  "severity": "1-Major Impact",
  "subject": "Subject",
  "description": "Test Description",
  "customer": "telefilm",
  "customerTrackingNumber": "TRACK-FSDJW452",
  "endCustomerName": "Customer 1",
  "partnerTicketNumber": "SDG232ASD23",
  "categoryType": "Technical",
```

```

"doNotShowInPortalFlag": false,
"questionnaire": {
  "questionnaireId": "100",
  "questions": [
    {
      "questionId": "670",
      "currentAnswers": [
        "2015-11-29T11:58:53.273Z"
      ]
    }
  ]
}

```

CasePost members

Member	Type	Required	Description
<p>CasePost: Encapsulates data necessary to create a new support ticket. Once created, much of this data is reflected back in the ticketInfo object.</p>			
additionalDetails	String	○	Provides additional text that describes the problem.
categoryType	String	✓	The ticket's relevant top-level category , either Billing , Business , Professional-Services , Security , or Technical .
customer	String	○	If a reseller creates a case for a third party, this specifies the company name for the account.
customerTrackingNumber	String	○	If a reseller creates a case on behalf of a third party, this allows the reseller its own way to uniquely identify the case ticket.
description	String	✓	An extended description of the problem. This field is ignored if a questionnaire object is present. In that case, leave the description blank and use additionalDetails to specify any descriptive text.
doNotShowInPortalFlag	Boolean	○	Whether the support ticket is suppressed from certain interfaces, false by default.
endCustomerName	String	○	If a reseller creates a case for a third party, this identifies the name of the end user for whom the ticket is being tracked.

partnerTicketNumber	String	○	Value provided by the user in limited cases when creating a new support ticket.
questionnaire	CasePost.questionnaire	○	Encapsulates questionnaire data the user answers prior to submitting a ticket. See the Questionnaire object.
severity	String	✓	Specifies the level of severity based on data available in CaseAttributes .
subCategories	CasePost.subCategories[]	✓	Specifies a set of Subcategory elements initially assigned to the support ticket.
subject	String	✓	Specifies a subject line that serves as a title for the support ticket.
userDetail	CasePost.userDetail	✓	Encapsulates data about the user on whose behalf the support ticket was created, which does not necessarily match who provisioned the API token.

CasePost.questionnaire: Encapsulates questionnaire data the user answers prior to submitting a ticket. See the [Questionnaire](#) object.

questionnaireId	String	✓	Unique identifier for the questionnaire.
questions	CasePost.questionnaire.questions[]	○	Lists questions the user has answered.

CasePost.questionnaire.questions[]: Lists questions the user has answered.

currentAnswers	Array	✓	The user's answers to the question, potentially more than one for multiple-choice questions.
questionId	String	✓	Unique identifier for each question.

CasePost.subCategories[]: Specifies a set of [Subcategory](#) elements initially assigned to the support ticket.

displayName	String	✓	Interface display text that corresponds to the subCategoryType .
		✓	

<code>subCategoryType</code>	String		The subcategory topic to assign to a new support ticket.
<code>subCategoryValue</code>	String	✓	The subcategory topic to assign to a new support ticket.
CasePost.userDetail: Encapsulates data about the user on whose behalf the support ticket was created, which does not necessarily match who provisioned the API token.			
<code>userCompany</code>	String	✓	Identifies the company on whose behalf the support request is created.
<code>userEmail</code>	String	✓	The user's email address.
<code>userName</code>	String	✓	The user's full name.
<code>userPhone</code>	String	✓	The user's telephone number.

Case

Encapsulates information about an individual support ticket. The object described here reflects a GET on a single support ticket. Collections feature an array of abbreviated [Case.ticketInfo](#) objects. For POST objects, see [CasePost](#).

Download schema: [Case.json](#)

Sample GET:

```
{
  "ticketInfo": {
    "caseId": "F-CS-1455428",
    "subject": "testing",
    "description": "testing",
    "status": "Unassigned",
    "severity": "3-Low Impact",
    "assignedTo": "In Queue",
    "categoryType": "Technical",
    "createdDate": "2016-02-29T14:15:40+0000",
    "modifiedDate": "2016-03-09T08:57:13+0000",
    "customer": "netflix",
    "externalSystemNumber": "AD-1432",
    "geoLinkId": "1-HGAJ",
    "partnerTicketNumber": "ADYH-1272",
    "endCustomerName": "a-netflix",
```

```
"nextStepExternal": "NA",
```

Case members

Member	Type	Required	Description
Case: Encapsulates information about an individual support ticket. The object described here reflects a GET on a single support ticket. Collections feature an array of abbreviated Case.ticketInfo objects. For POST objects, see CasePost .			
activities	Case.activities[]	○	Specifies user-initiated actions after a support ticket is created, such as adding comments or uploading files.
caseMilestones	Case.caseMilestones[]	✓	A read-only list of state changes generated as the support ticket progresses.
permissions	Case.permissions	✓	Represents types of follow-up activities the support ticket's submitter is allowed.
solution	Case.solution	✓	Read-only Akamai-supplied details about how a support ticket is resolved.
ticketInfo	Case.ticketInfo	✓	Encapsulates high-level data about the support ticket, including information that appears within Case listings, and information initially supplied when creating the ticket.
userDetail	Case.userDetail	✓	Encapsulates data about the user on whose behalf the support ticket was created, not necessarily the same as the submittedBy user who created the ticket.
Case.activities[]: Specifies user-initiated actions after a support ticket is created, such as adding comments or uploading files.			
activityCreatedBy	String	○	The name of the user following up on the support ticket.
activityCreatedDate	String	✓	An ISO 8601-formatted timestamp for when the follow-up activity occurred.
activityDescription	String	✓	An extended description, which corresponds to update requests' comment member.
		✓	

<code>activitySubject</code>	String		Specifies a subject line that serves as a title.
<code>activityType</code>	Enumeration	✓	Characterizes the activity, either <code>email</code> , <code>attachment</code> , or <code>webUpdate</code> .
<code>fileName</code>	String	○	If the activity is a file upload, this specifies the filename.
<code>links</code>	Hypermedia array	○	If the activity is a file upload, locates the URI path to download it.
Case.caseMilestones []: A read-only list of state changes generated as the support ticket progresses.			
<code>caseMilestoneDate</code>	String	○	An ISO 8601-formatted timestamp identifying when the state change occurred.
<code>caseMilestoneId</code>	Enumeration	✓	A status message generated during a support ticket's lifetime, either <code>created</code> , <code>pending</code> , <code>initialResponse</code> , <code>assigned</code> , <code>inProgress</code> , or <code>closed</code> . Each state may only occur once over the lifetime of the ticket. Note that these values do not match the ticket's set of potential <code>status</code> values.
Case.permissions: Represents types of follow-up activities the support ticket's submitter is allowed.			
<code>addNote</code>	Boolean	✓	Whether the user can append notes to the ticket.
<code>requestClose</code>	Boolean	✓	Whether the user can request that the ticket be closed.
<code>uploadFile</code>	Boolean	✓	Whether the user can upload file attachments as part of the support ticket.
Case.solution: Read-only Akamai-supplied details about how a support ticket is resolved.			
<code>investigation</code>	String	○	Identifies the nature of the problem.
<code>resolution</code>	String	○	Characterizes the solution to the problem.
<code>solutionSubject</code>	String	✓	Provides a subject line that serves as a title.

Case.ticketInfo: Encapsulates high-level data about the support ticket, including information that appears within Case listings, and information initially supplied when creating the ticket.

assignedTo	String	✓	The group within Akamai assigned to the support ticket. The value may be In Queue prior to assignment.
caseId	String	✓	Unique identifier for each support ticket.
categoryType	Enumeration	✓	The ticket's relevant top-level category , either Billing , Business , Professional-Services , Security , or Technical .
createdDate	String	✓	The ISO 8601-formatted time when the case was first created.
customer	String	○	If a reseller creates a case for a third party, this specifies the company name for the account.
customerTracking Number	String	○	If a reseller creates a case on behalf of a third party, this allows the reseller its own way to uniquely identify the case ticket.
description	String	✓	An extended description of the problem.
endCustomerName	String	○	If a reseller creates a case for a third party, this identifies the name of the end user for whom the ticket is being tracked.
externalSystem Number	String	○	Value provided by the user in limited cases when creating a new support ticket.
geoLinkId	String	○	Generated value in limited cases for new support tickets.
modifiedDate	String	✓	The ISO 8601-formatted time when the case was last modified. This encompasses any change to the overall object, not merely within the scope of the Case.ticketInfo sub-object in which the member appears.
nextStepExternal	String	○	Indicates the next steps the case's assignee

are to take.

<code>partnerTicketNumber</code>	String	○	Value provided by the user in limited cases when creating a new support ticket.
<code>severity</code>	String	✓	Specifies the level of severity based on data available in CaseAttributes .
<code>status</code>	String	✓	The current status of the support ticket. Note that these values do not match the set of potential <code>caseMilestoneId</code> values.
<code>subCategories</code>	Case.ticketInfo.subCategories[]	✓	Specifies a set of subcategories initially assigned to the support ticket.
<code>subject</code>	String	✓	Specifies a subject line that serves as a title for the support ticket.
<code>submittedBy</code>	String	✓	The name of the user associated with the API token accessing the case ticket, not necessarily the same as the <code>userName</code> on whose behalf the ticket is filed.

`Case.ticketInfo.subCategories []`: Specifies a set of subcategories initially assigned to the support ticket.

<code>displayName</code>	String	✓	Interface display text that corresponds to the <code>subCategoryType</code> .
<code>subCategoryType</code>	String	✓	The subcategory topic to assign to a new support ticket.
<code>subCategoryValue</code>	String	✓	The subcategory topic to assign to a new support ticket.

`Case.userDetail`: Encapsulates data about the user on whose behalf the support ticket was created, not necessarily the same as the `submittedBy` user who created the ticket.

<code>userCompany</code>	String	✓	Identifies the company on whose behalf the support request is created.
<code>userEmail</code>	String	✓	The user's email address.
		✓	

<code>userName</code>	String		The user's full name.
<code>userPhone</code>	String	✓	The user's telephone number.

Response

Common response object when adding a note or file upload, or when closing a case.

Download schema: [Response.json](#)

Sample GET:

```
{
  "caseId": "F-CS-1203600",
  "statusMessage": "Ticket has been successfully updated with notes",
  "links": [
    {
      "href": "/case-management/v2/cases/F-CS-1203600",
      "rel": "parent"
    }
  ]
}
```

Response members

Member	Type	Required	Description
Response	Common response object when adding a note or file upload, or when closing a case.		
<code>caseId</code>	String	✓	Unique identifier for the support ticket.
<code>links</code>	Hypermedia array	✓	Locates the resource for the <code>caseId</code> .
<code>statusMessage</code>	String	✓	A status message for the request.

Errors

This section provides details on the data object that reflects the API's common response to error cases, and lists the API's range of response status codes for both error and success cases.

Error responses

The following shows a typical error response. The outer object characterizes the overall problem, while

the `details` array lists potentially more than one problem detected in the request.

```
{
  "code": "bad.request",
  "incidentId": "0c296a9e-b063-44d1-a6c0-35228becc4ab",
  "title": "Bad Request",
  "details": [
    {
      "code": "bad.request.error",
      "message": "CaseId provided is incorrect"
    }
  ]
}
```

Error members

Member	Type	Description
Required		
<code>code</code>	String	Identifies a type of error that characterizes the overall problem.
<code>details</code>	Error.details	Provides details on potentially more than one error detected in the request.
<code>incidentId</code>	String	Provides a unique identifier for the problem, useful when communicating with Akamai support.
<code>title</code>	String	A descriptive label for the overall problem.

Error.details

Member	Type	Description
Required		
<code>code</code>	String	Identifies a type of error that characterizes each specific problem.
<code>message</code>	String	Describes each specific problem.

HTTP status codes

The following lists the range of HTTP response codes the API may produce for both success and error cases:

Code	Description
200	Request OK
201	Resource created
401	Unauthorized request
402	Failed request
403	Forbidden
404	Resource not found
405	Method not allowed
415	Unsupported media type
429	Too many requests
500	Internal Server Error
503	Service Unavailable

COMPANY

[Akamai.com](#)
[Akamai Locations](#)
[Contact Support](#)

STAY IN TOUCH

DEVELOPER

[Documentation](#)
[Blog](#)
[Resources](#)

