

API Methods: createCheck Endpoint



CHECKCENTRAL



BINARYFORTRESS

API Methods: createCheck Endpoint

The CheckCentral API provides an endpoint for programmatically creating Checks.

To create a new Check through the API, an API token with Read/Write access is required. Organization administrators can create tokens through the API portal on your dashboard.

The screenshot displays the CheckCentral dashboard interface. At the top, a navigation bar includes links for Dashboards, Checks, Notifications, Activities, Organization, and Help. The Organization dropdown menu is open, showing options like API, Ignore Rules, Settings, Stats, Subscription, Users, and User Groups. The API option is highlighted with a green box. Below the navigation bar, the main content area features the Binary Fortress logo and a section titled 'API Tokens'. This section includes a '+ Add Token' button (highlighted with a green box) and a 'Documentation' link. Below this, there is a large grey box with the text 'no api tokens found'. A similar section for 'User API Tokens' is visible below, also with a '+ Add Token' button and a 'Documentation' link, followed by a grey box with the text 'no api tokens found'.

"User API Tokens" will only return data available to the user based on their granted permissions and user groups. Regular "API Tokens" have the Access Level given upon creation.

The endpoint is located at `https://api.checkcentral.cc/createCheck/?apiToken=APITOKEN` where the APITOKEN placeholder is replaced with your valid token.

The request must be made with the `Content-Type` header set to `application/json`. The body of the request should contain as many of the properties below to properly define the Check. Any properties not passed will be set to their default value.

For example, we want to create a new Check that:

- Receives messages at the address `demo+apicheck@mycheckcentral.cc`.
- Expects messages every two hours.
- Expects messages only on weekdays.
- Is set to `Failure` unless the message subject is `Message Success` or the message body has the text `No Errors Reported`.

To achieve this, we send the following json in the body of the request:

```
{
  "name": "An API Check",
  "email_alias": "apicheck",
  "interval_type": "hour",
  "interval_value": 2,
  "active_days": [
    "mon",
    "tue",
    "wed",
    "thu",
    "fri"
  ],
  "success_conditions": {
    "required_conditions": "any",
    "conditions": [
      {
        "message_field": "subject",
        "matches": "exactly",
        "value": "Message Success"
      },
      {
        "message_field": "body",
        "matches": "contains",
        "value": "No Errors Reported"
      }
    ]
  }
}
```

The check data must include a `name` property, as well as at least one `email_alias` and `matching_conditions` to match the Check against incoming messages.

The complete list of possible properties for check configuration are as follows:

Check Structure

Parameter	Type/Allowed Values	Default
<code>name</code> The name for the new check. This parameter is required.	String	
<code>email_alias</code> The 'plus alias' for the check. This will be used to generate the custom email address for the Check along with your organization account name (account_name+email_alias@mycheckcentral.cc). At least one <code>email_alias</code> or <code>matching_conditions</code> parameter must be specified.	String	
<code>matching_conditions</code> The rules that will be used to match incoming messages. (See the ConditionGroup table below.) At least one <code>email_alias</code> or <code>matching_conditions</code> parameter must be specified.	ConditionGroup	<code>{}</code>
<code>group_id</code> The ID of the group to assign this check. (IDs may be retrieved from the getGroups API endpoint.)	String	
<code>description</code> A text description for the check.	String	

<div data-bbox="64 237 264 289" data-label="Text"><code>interval_type</code></div> <p>The interval unit CheckCentral will use to calculate the expected arrival time of incoming messages.</p>	<div data-bbox="1024 153 1114 184" data-label="Text">One of:</div> <div data-bbox="1045 237 1502 499" data-label="List-Group"> <ul style="list-style-type: none"> minute hour day week month <div data-bbox="1442 300 1502 352" data-label="Text">day</div> </div>
<div data-bbox="64 573 279 625" data-label="Text"><code>interval_value</code></div> <p>The number of interval units (see <code>interval_type</code> above) used when calculating the expected arrival of incoming messages.</p>	<div data-bbox="1024 646 1114 678" data-label="Text">Integer</div> <div data-bbox="1442 636 1474 688" data-label="Text">1</div>
<div data-bbox="64 835 292 888" data-label="Text"><code>overdue_minutes</code></div> <p>How many minutes after the interval has elapsed CheckCentral will wait before flagging the check as failed.</p>	<div data-bbox="1024 909 1114 940" data-label="Text">Integer</div> <div data-bbox="1442 898 1482 951" data-label="Text">30</div>
<div data-bbox="64 1087 251 1140" data-label="Text"><code>window_start</code></div> <p>An optional time of day when messages will begin to be received.</p>	<div data-bbox="1024 1140 1097 1171" data-label="Text">String</div> <div data-bbox="1442 1150 1531 1203" data-label="Text">00:00</div>
<div data-bbox="64 1297 224 1350" data-label="Text"><code>window_end</code></div> <p>An optional time of day when messages will cease to be received.</p>	<div data-bbox="1024 1350 1097 1381" data-label="Text">String</div> <div data-bbox="1442 1360 1531 1413" data-label="Text">00:00</div>
<div data-bbox="64 1675 235 1728" data-label="Text"><code>active_days</code></div> <p>Pass an array of days to indicate which days the check should expect to receive messages.</p>	<div data-bbox="1024 1518 1417 1602" data-label="Text">An array containing either <code>all</code>, or any of:</div> <div data-bbox="1045 1654 1502 2001" data-label="List-Group"> <ul style="list-style-type: none"> sun mon tue wed thu fri sat <div data-bbox="1442 1738 1502 1791" data-label="Text">all</div> </div>

`enabled_notifications`

Pass an array of personal notification services to configure the check to send alerts via those services.

An array containing any of:

- email
 - sms
 - push
 - pushbullet
 - pushover
- `[]`

`enabled_integrations`

Pass an array of organization notification services to configure the check to send alerts via those services.

An array containing any of:

- slack
 - mattermostwebhook
 - discordwebhook
 - googlechatwebhook
 - slackwebhook
 - microsoftteamswebhook
 - customwebhook
 - jirasoftwarewebhook
 - zendeskapi
 - ifttt
 - webexwebhook
 - zapierwebhook
 - rocketchatwebhook
 - zulipwebhook
 - asanaapi
 - externalemails
 - redmineapi
 - bugzillaapi
 - githubapi
 - gitlabapi
 - microsofttodolistapi
- `[]`

`notify_returned_to_success`

Checks with this flag will send an alert when they are restored to a success state by an incoming message.

Bool

`true`

`notify_outside_window`

Checks with this flag set to false will only send alerts during the time window and active days configured above.

Bool

`true``notify_consecutive_alerts`

This parameter controls when CheckCentral will send alerts about repeated failure or warning messages.

One of:

- `always`
- `first`
- `after_n`
- `every_nth`

`always``notify_consecutive_alerts_limit`

If `notify_consecutive_alerts` is set to either `after_n` or `every_nth`, this parameter configures how many consecutive failures or consecutive warnings will trigger alerts.

Integer

`1``notify_failure_grace_period`

The number of minutes this check will wait to send alerts about failures. If the check is returned to successful status before this period elapses, any pending alerts will be cancelled.

Integer

`0``default_status`

The default status for incoming messages that don't match against any of the conditions below.

One of:

- `success`
- `warning`
- `failure`

`failure``success_conditions`

The rules that determine whether incoming messages will be set to success. See the definition of a [ConditionGroup](#) below.

ConditionGroup

`{}`

warning_conditions

ConditionGroup

{}

The rules that determine whether incoming messages will be set to warning.
See the definition of a [ConditionGroup](#) below.

failure_conditions

ConditionGroup

{}

The rules that determine whether incoming messages will be set to failure.
See the definition of a [ConditionGroup](#) below.

ConditionGroup

Parameter	Type/Allowed Values	Default
<p>required_conditions</p> <p>This parameter controls whether incoming messages must meet any or all of the conditions defined below to be considered a match.</p>	<p>One of:</p> <ul style="list-style-type: none">allany	<p>all</p>
<p>condense_whitespace</p> <p>If this flag is set, all of the conditions defined in this group will treat multiple whitespace characters as a single space.</p>	<p>Bool</p>	<p>false</p>
<p>conditions</p> <p>The list of conditions for this ConditionGroup to match messages against. See the definition of a Condition below.</p>	<p>An array containing one or more of: [Condition]</p>	<p>{}</p>

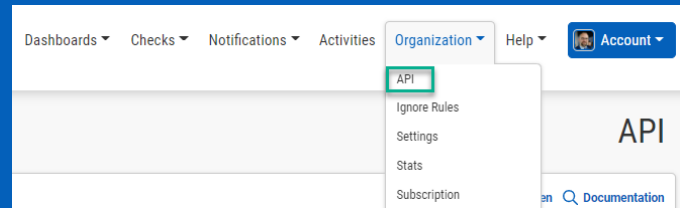
Condition

Parameter	Type/Allowed Values	Default
<div>message_field</div> <p>Which part of the incoming message will be examined for this condition.</p>	<p>One of:</p> <ul style="list-style-type: none">subjectbodytoccattachmentsattachment_textfromoriginally_to	<div>subject</div>
<div>matches</div> <p>What matching rule this condition will use.</p>	<p>One of:</p> <ul style="list-style-type: none">containsnot_containsexactlyemptynot_emptyhas_attachmentshas_no_attachmentsfilename_containsfilename_not_containsfilename_exacttext_containstext_not_containscomplex_matchcomplex_filename_matchcomplex_text_match	<div>contains</div>
<div>value</div> <p>The value to use with the above matching rule and message component.</p>	String	

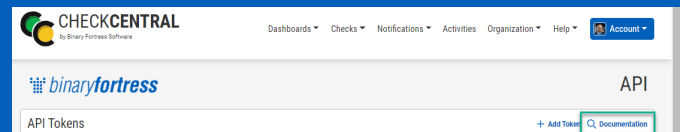
More API Documentation and Help

A full list of API Methods are available via the API page under your CheckCentral organization.

Log in to your account. Click "Organization" from the main menu, then "API."



Click "Documentation."



Here you'll see the list of Methods available, their permission requirements, their purpose, and the call limit per day.

CheckCentral API			Return to API Tokens	Contact Support
Method	Description	Calls Per Day		
Read Only API Token Required				
getChecks	Gets information about the checks for an organization.	15/10000		
getActivities	Gets Activities.	0/10000		
getCheckGroups	Gets information about the groups for an organization.	0/10000		
getDashboards	Gets information about the custom dashboards for an organization.	0/10000		
getIssues	Gets a list of current issues for an organization.	0/10000		
getOverview	Reports on the overall status of an organization.	0/10000		
getPermissions	Gets a list of permissions for a check group, user, or user group.	0/10000		
generateReport	Serve the specified report data.	0/10000		
getReports	Gets a list of reports for an organization.	0/10000		
getUsers	Gets a list of users for an organization.	0/10000		
getUserGroups	Gets a list of user groups for an organization.	0/10000		
Read/Write API Token Required				
createActivities	Create new activities.	0/10000		

Click on any of the Methods to see a testing area with parameter requirements and an example of usage. Populate the fields and click "Send Test Request" to see the JSON Response.

getChecks					Return to API Documentation	Contact Support
Parameter	Value	Default Value	Data Type	Description		
apiToken	<input type="text"/>	-	string	The API token used to authenticate your organization. API token with Read Only access required.	<input checked="" type="checkbox"/>	
checkid	<input type="text"/>	-	string	Pass a check ID to return only a single check.	<input type="checkbox"/>	
checkName	<input type="text"/>	-	string	Pass a value to return checks with names that contain that value. This will be overridden by the checkid parameter.	<input type="checkbox"/>	
activityCount	<input type="text" value="0"/>	0	int	Include the recent activities of the returned checks. The response is limited to a maximum of 30 activities per check.	<input type="checkbox"/>	
includeSettings	<input type="checkbox"/>	false	bool	Include the configured settings of returned checks.	<input type="checkbox"/>	
<button>Send Test Request</button>						
<h3>How to Use This API</h3>						
To use this API you need to send a GET request to the endpoint shown below. This endpoint is limited to 10,000 calls per day.						
Endpoint: https://api.checkcentral.cc/getChecks/?apiToken=						
Curl Example: <code>curl -X GET "https://api.checkcentral.cc/getChecks/?apiToken="</code>						
If you need any help using this API, please feel free to contact us .						
<h3>JSON Response</h3>						
Send a test request to see what the JSON response would look like.						

About CheckCentral

CheckCentral Monitoring consolidates and simplifies backup, system, and software email updates into a clean, graphical dashboard, bringing peace of mind to IT administrators of SMBs, Enterprises, and MSPs.

To learn more about CheckCentral, visit: <https://www.checkcentral.cc>

About Binary Fortress Software

Binary Fortress has spent 18 years in pursuit of one goal: create software to make life easier. Our software ranges from display management and system enhancement utilities to monitoring tools and digital signage. IT administrators, professional gamers, coffee-shop owners, and MSPs all rely on Binary Fortress to make their days better, and their lives easier.

Copyright © 2007-2025 Binary Fortress Software, all rights reserved.

The Binary Fortress logo is a trademark of Binary Fortress Software.

The CheckCentral logo is a trademark of Binary Fortress Software.

Binary Fortress Software
1000 Innovation Drive, Suite 500
Kanata, Ontario, Canada
K2K3E7
<https://www.binaryfortress.com>