



# Import and Export Health Rule Configurations

## On this page:

- [Exporting Health Rules from an Application](#)
- [Importing Health Rules to an Application](#)

## Related pages:

- [Configure Health Rules](#)
- [Import and Export Transaction Detection Configuration](#)
- [Use Curl to Access AppDynamics REST APIs](#)

You can export your health configurations from one application to another using a special AppDynamics REST API. This capability allows you to re-use health rule configurations in different applications instead of re-configuring each application manually from the AppDynamics console.

If you can view a health rule, you can export it.

If you can create a health rule, you can import it.

## Exporting Health Rules from an Application

Exports are HTTP GET operations.

### Export the configurations for all health rules in an application

`http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>`

#### Example

```
http://pml.appdynamics.com:80/controller/healthrules/3
```

produces the output in [all\\_health\\_rules](#).

### Export the configuration for a single health rule

`http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>?name=<health_rule_name>`

For example:

```
http://pml.appdynamics.com/controller/healthrules/3?name=Business Transaction response time is much higher than normal
```

produces the output in [one\\_health\\_rule](#).

## Importing Health Rules to an Application

Imports are HTTP POST operations.

After you have exported health rules you can import them to a different application passing the xml file created by the export operation as payload to the POST. You can modify the exported file before you import it. You might want to do this to add or remove one or more health rule configurations or to change their names.

Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

The default behavior is not to overwrite an existing health rule of the same name. If you want to overwrite an existing health rule of the same name, specify the **overwrite=true** parameter. since the default is **false**.

The syntax is the same for importing one health rule configuration or several. All the health rule configurations in the xml file are imported.

## Import the configurations for health rules in an application

http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>?overwrite=true|false

This example imports the health rules in the uploaded file, overwriting any health rules of the same name.

The screenshot shows a web interface for importing health rules. The URL bar contains `http://pm1.appdynamics.com:80/controller/healthrules/3?overwrite=true`. Below the URL, there are radio buttons for HTTP methods: GET, POST (selected), PUT, PATCH, DELETE, HEAD, OPTIONS, and Other. There are three tabs: Raw, Form, and Headers. Below these tabs, there is a section for the payload. It has tabs for Raw, Form, Files (1), and Payload. Under the Files (1) tab, there is a link "Add new file field". Below that, there is a "Choose Files" button, a text input field containing "all\_health\_rules", a "fileUpload" button, and a close button "x". To the right of the file input, it says "all\_health\_rules (30.8 KB)". At the bottom, there is a dropdown menu showing "application/xml" and a note: "Set 'Content-Type' header to overwrite this value."

The next example imports the health rules without overwriting. In this case, any health rules in the destination controller that have the same names as health rules in the all\_health\_rules file are not overwritten.

▶ http://pm1.appdynamics.com:80/controller/healthrules/3

GET  POST  PUT  PATCH  DELETE  HEAD  OPTIONS  Other

**Raw** **Form** **Headers**

---

**Raw** **Form** **Files (1)** **Payload**

[Add new file field](#)

all\_health\_rules   all\_health\_rules (30.8 KB)

*Set "Content-Type" header to overwrite this value.*