Configuring the Uptime Controller

Securing the Controller

If you plan on using the Uptime Controller by integrating API functions with the Uptime Infrastructure Monitor Web interface, consider doing the following to prevent several common browser-related warnings when navigating secure and non-secure pages within the same web page:

- Enable SSL for the Uptime Infrastructure Monitor Web interface, using the instructions found in Implementing HTTPS Browsing for Web Interface with Apache 2.2 or Implementing HTTPS Browsing for the Web Interface with Apache 2.4.
- Purchase a valid SSL certificate for the Uptime Infrastructure Monitor Web interface to avoid warnings about a self-signed certificate in the browser.
  The Controller installs an unsigned and unverified certificate as part of its installation process. For installations in a production environment, a valid key should be purchased and installed on the Controller.

Reflecting Configuration Changes in the Proxy

As part of the Uptime Infrastructure Monitor installation process, a proxy is configured in the `<uptimeDirectory>\apache\conf\httpd.conf` file. This provides ease of use with the Uptime Controller, as it simplifies many API programming tasks, and helps to ensure the Controller remains secure.

By default the proxy section should look like the following:

```
# proxy settings
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
ProxyRequests Off
<Proxy *>
  Order deny,allow
  Allow from all
</Proxy>
ProxyPass /uptime http://<uptimeHost>:9996/uptime retry=0
ProxyPassReverse /uptime http://<uptimeHost>:9996/uptime
ProxyPass /gadgets/service http://<uptimeHost>:9996/gadgets/service retry=0
ProxyPassReverse /gadgets/service http://<uptimeHost>:9996/gadgets/service
ProxyPass /gadgets/instances http://<uptimeHost>:9996/gadgets/instances retry=0
ProxyPassReverse /gadgets/instances http://<uptimeHost>:9996/gadgets/instances

# ssl proxy settings
LoadModule ssl_module modules/mod_ssl.so
SSLProxyEngine on
ProxyPass /api https://<controllerHost>:9997/api retry=0
ProxyPassReverse /api https://<controllerHost>:9997/api
```

The proxy is configured to work using the default options and configuration choices made when Uptime Infrastructure Monitor is installed:

- the Monitoring Station (`<uptimeHost>` above) and Uptime Controller (`<controllerHost>` above) are part of the same installation (i.e., they have the same value), and assumed to be on the same host
- the port the Controller uses to communicate with Uptime Infrastructure Monitor is `9997`

If there are changes to the way Uptime Infrastructure Monitor is deployed on the network, you need to modify the Apache Web server configuration file, accordingly:

Deployment Change	Affected Proxy Setting
change the API hostname or move the API to another server	`<controllerHost>`
change the API port	`9997`
when on running on different servers, the Monitoring Station's network location is changed	`<uptimeHost>`

After making any change, restart the `Uptime Web Server` service (on the correct Uptime Infrastructure Monitor instance, if applicable) to apply these changes. Then verify that you correctly configured the proxy by browsing to `https://<uptimeMonitoringStation>/api`. The behavior should be identical to browsing to the Uptime Controller at `https://<uptimeController>/api`.

Controller Configuration Parameters

The following list details several common Uptime Controller configuration tasks.

Task	Setting	Configuration Location in `<uptimeDirectory>\controller\`	Notes

change Controller port	Uptime Controller port	`\etc\jetty-ssl.xml`	To change the port, find this line within `jetty-ssl.xml`: `<Set name="Port"><Property name="jetty.port" default="9997"/></Set>` Update the `9997` value to an unassigned port, and then restart the Uptime Controller service.

change Controller DataStore target	database connection settings	`\resources\uptime-controller.conf`	Example connection fields and options are provided for all supported DataStore back-end platforms. By default, the Uptime Controller uses a standard MySQL DataStore connection (unless you updated these values during installation).
			
			`dbType=MYSQL`
			`dbHostname=localhost`
			`dbPort=3308`
			`dbName=uptime`
			`dbUsername=uptime`
			`dbPassword=uptime`
			`dbJdbcProperties=`
			
			To change the connection details, update these fields with your database platform information. There are example entries in the configuration file.
			
			After you have updated your settings, restart the Uptime Controller service.

indicate where PHP sessions are stored	PHP session directory	`\resources\uptime-controller.conf`	By default, the `phpSessionDirectory` parameter is not defined, and Uptime Infrastructure Monitor looks for stored sessions in the Apache work space: • `<uptimeDirectory>\apache\tmp` (Windows) • `/usr/local/uptime/apache/tmp` (Linux)
change Controller logging level	logging level	`\resources\uptime-controller.conf`	The Controller automatically logs events in the `/logs` directory. Logging levels include the following: • `TRACE` • `DEBUG` • `INFO` • `WARN` • `ERROR` • `FATAL` By default, the logging level is `INFO`. After you have updated your settings, restart the Uptime Controller service.
install new Controller SSL certificate	SSL communication	`\etc\jetty-ssl.xml`	To install a new key, follow steps 3 and 4 from Configuring SSL/TLS, and then restart the Uptime Controller service.

configure SSL /TLS parameters	SslContext Factory	`\etc\jetty-ssl.xml`	`serverIncludeCipherSuites=ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384`
			`serverIncludeProtocols=TLSv1.2`
			`# serverExcludeCipherSuites=`
			`serverExcludeProtocols=SSLv3`