

CSP Server Configuration API

Version 1.0.1

Paths

/v1/updates/{cspId}

GET /v1/updates/{cspId}

Description

Retrieves a list of available updates, for registered modules of the CSP. See the "register" API call on how to register modules

Parameters

Name	Located in	Description	Required	Schema
cspId	path	a unique identifier that defines a Registered and Known CSP. The csp identifier follows the UUID formatted as text, for 36 characters total, arranged as 8-4-4-4-12. Example = 123e4567-e89b-12d3-a456-426655440000	Yes	⇒ string

Responses

Code	Description	Schema
200	List of available updates by module	⇒ ▼ UpdateInformation { contains update information for the selected CSP dateChanged: ▶ string * available: ▶ { } }
404	cspId not found; failure to identify cspId presented	⇒ ▼ ResponseError { Failed Response object responseCode: ▶ integer * responseText: ▶ string * responseException: string * }
500	Systemic failure	⇒ ▼ ResponseError { Failed Response object responseCode: ▶ integer * responseText: ▶ string * responseException: string * }

Try this operation

/v1/appInfo/{cspId}

Description

Submits a body that contains information of the CSP

Parameters

Name	Located in	Description	Required	Schema
cspId	path	a unique identifier that defines a Registered and Known CSP. The csp identifier follows the UUID formatted as text, for 36 characters total, arranged as 8-4-4-4-12. Example = 123e4567-e89b-12d3-a456-426655440000	Yes	\Rightarrow string
heartbeatInfo	body	the information	Yes	\Rightarrow <pre> ▼ AppInfo { AppInfo object name: ▶ string * recordDateTime: ▶ string * moduleInfo: ▶ ModulesInfo { } } </pre>

Responses

Code	Description	Schema
200	Successful response	\Rightarrow <pre> ▼ Response { Accepted Response object responseCode: ▶ integer * responseText: ▶ string * } </pre>
404	cspId not found; failure to identify cspId presented	\Rightarrow <pre> ▼ ResponseError { Failed Response object responseCode: ▶ integer * responseText: ▶ string * responseException: string * } </pre>

Code	Description	Schema
500	Systemic failure	<div><div>↔</div><div><div>▼ResponseError { Failed Response object responseCode: integer * responseText: string * responseException: string * }</div></div></div>
<div>Try this operation</div>		

/v1/register/{cspId}

Description

Register a NEW csp or register for an existing CSP the modules that are being installed

Parameters

Name	Located in	Description	Required	Schema
cspId	path	a unique identifier that defines a Registered and Known CSP. The csp identifier follows the UUID formatted as 8-4-4-4-12. Example = 123e4567-e89b-12d3-a456-426655440000	Yes	⇒ string
cspRegistration	body	a block of information to register the CSP being installed	Yes	⇒ <div><div>▼Registration { Registration object for a new CSP name: ▶ string registrationDate: ▶ string externalIPs: ▶ [] internalIPs: ▶ [] registrationIsUpdate: ▶ boolean * contacts: ▶ [] moduleInfo: ▶ ModulesInfo { } }</div></div>

Responses

Code	Description	Schema
------	-------------	--------

200	received when the CSP is registered successfully	⇒	<div>▼Response {</div> <div>Accepted Response object</div> <div>responseCode: ▶ integer *</div> <div>responseText: ▶ string *</div> <div>}</div>
400	received when CSP updates entries with "registrationIsUpdate = false"	⇒	<div>▼ResponseError {</div> <div>Failed Response object</div> <div>responseCode: ▶ integer *</div> <div>responseText: ▶ string *</div> <div>responseException: string *</div> <div>}</div>
404	received when CSP requests update but original entry does not exist OR when cspld is not recognized	⇒	<div>▼ResponseError {</div> <div>Failed Response object</div> <div>responseCode: ▶ integer *</div> <div>responseText: ▶ string *</div> <div>responseException: string *</div> <div>}</div>
500	received when the CSP is not registered due to system error	⇒	<div>▼ResponseError {</div> <div>Failed Response object</div> <div>responseCode: ▶ integer *</div> <div>responseText: ▶ string *</div> <div>responseException: string *</div> <div>}</div>

Try this operation

Models

VersionDefinition

- ▼**VersionDefinition** integer minimum:10000 maximum:90000
- ⇒ the version definition used for communicating with the CSP APIs; it has a visual and a numeric format. The visual format can be created by splitting the numeric at specific points, (0 based) 1, 2 example: 12000 split as 1.2.000

Registration

```

▼Registration {
  Registration object for a new CSP
  name:           ▶ string *
  registrationDate: ▶ string *
  externalIPs:    ▶ []
  ⇒ internalIPs:   ▶ []
  registrationIsUpdate: ▶ boolean *
  contacts:       ▶ []
  moduleInfo:     ▶ModulesInfo { }
}

```

ContactDetails

```

▼ContactDetails {
  personName: string *
  ⇒ personEmail: ▶ string *
  contactType: ▶ string *
}

```

AppInfo

```

▼AppInfo {
  AppInfo object
  name:           ▶ string *
  ⇒ recordDateTime: ▶ string *
  moduleInfo:     ▶ModulesInfo { }
}

```

Response

```

▼Response {
  Accepted Response object
  ⇒ responseCode: ▶ integer *
  responseText: ▶ string *
}

```

ResponseError

```

▼ResponseError {
  Failed Response object
  responseCode: ▶ integer *
  ⇒ responseText: ▶ string *
  responseException: string *
}

```

ModulesInfo

```

▼ModulesInfo {
  The json object that contains information on the modules installed
  ⇒ modules: ▶[]
  data: ▶ { }
}

```

ModuleData

```

▼ModuleData {
  Map of data specific to a module
  fullName:    string *
  version:     ►VersionDefinition integer *
⇒ installedOn: ► string *
  active:      boolean *
  hash:        string *
}

```

UpdateInformation

```

▼UpdateInformation {
  contains update information for the selected CSP
⇒ dateChanged: ► string *
  available:    ► { }
}

```

ModuleUpdateInfo

```

▼ModuleUpdateInfo {
  List of available updates for a module
  name:          string *
  description:    string *
⇒ version:       ►VersionDefinition integer *
  released:      ► string *
  isIncremental: ► boolean *
  hash:          ► string *
}

```