

restClientConfig object description

When connection is established, JSON object `restClientConfig` is delivered in 200 OK response to the '`connect`' method. This object defines behavior for all subsequent calls concerning error handling, parameters overwriting and set of transmitted fields forming. If `restClientConfig` was not delivered, default values are used.

The below table lists all fields of the `restClientConfig` object and how the WCS server interprets them.

Parameter	Description	Default value						
clientExclude	The list of fields to exclude from sending to a client (comma separated)	empty						
restExclude	The list of fields to exclude from sending to REST (comma separated)	empty						
restOnError	Reaction to errors or 4xx statuses: <table><tr><td>LOG</td><td>Just log</td></tr><tr><td>FAIL</td><td>React and send the error</td></tr></table>	LOG	Just log	FAIL	React and send the error	LOG		
LOG	Just log							
FAIL	React and send the error							
restPolicy	Ability to redefine fields: OVERWRITE , NOTIFY or DISABLE <table><tr><td>OVERWRITE</td><td>Overwrite</td></tr><tr><td>NOTIFY</td><td>Do not overwrite</td></tr><tr><td>DISABLE</td><td>Do not send to REST</td></tr></table>	OVERWRITE	Overwrite	NOTIFY	Do not overwrite	DISABLE	Do not send to REST	NOTIFY
OVERWRITE	Overwrite							
NOTIFY	Do not overwrite							
DISABLE	Do not send to REST							
restOverwrite	The list of fields to overwrite if the option restPolicy= OVERWRITE (comma separated)	empty						

This is `restClientConfig` example. This object contains only '`ConnectionStatusEvent`' method. Other methods can be described same way:

```
"restClientConfig":
{
  "ConnectionStatusEvent":
  {
    "restExclude": "",
    "clientExclude": "",
    "restOnError": "LOG",
    "restPolicy": "NOTIFY",
    "restOverwrite": ""
  }
}
```

Example interpretation:

- These options are defined for `ConnectionStatusEvent` method: `restExclude`, `clientExclude`, `restOnError`, `restPolicy`, `restOverwrite`.
- When the method is called, all fields will be delivered to the web server because `restExclude=""`.
- When the method is called, all fields will be delivered to the client because `clientExclude=""`.
- If error occurs when accessing to the web server, this error will be logged but will not be delivered to the client. In this case, the client receives `normalConnectionStatusEvent` event.
- Even if the web server returns new parameters, these parameters will not be overwritten because `restPolicy="NOTIFY"`.
- `restOverwrite` field will not work because `restPolicy="NOTIFY"`.

`restClientConfig` object can describe another calls configuration such as: `call`, `OnCallEvent`, `CallStatusEvent`, `sendMessage`, `OnMessageEvent`, `MessageStatusEvent` etc