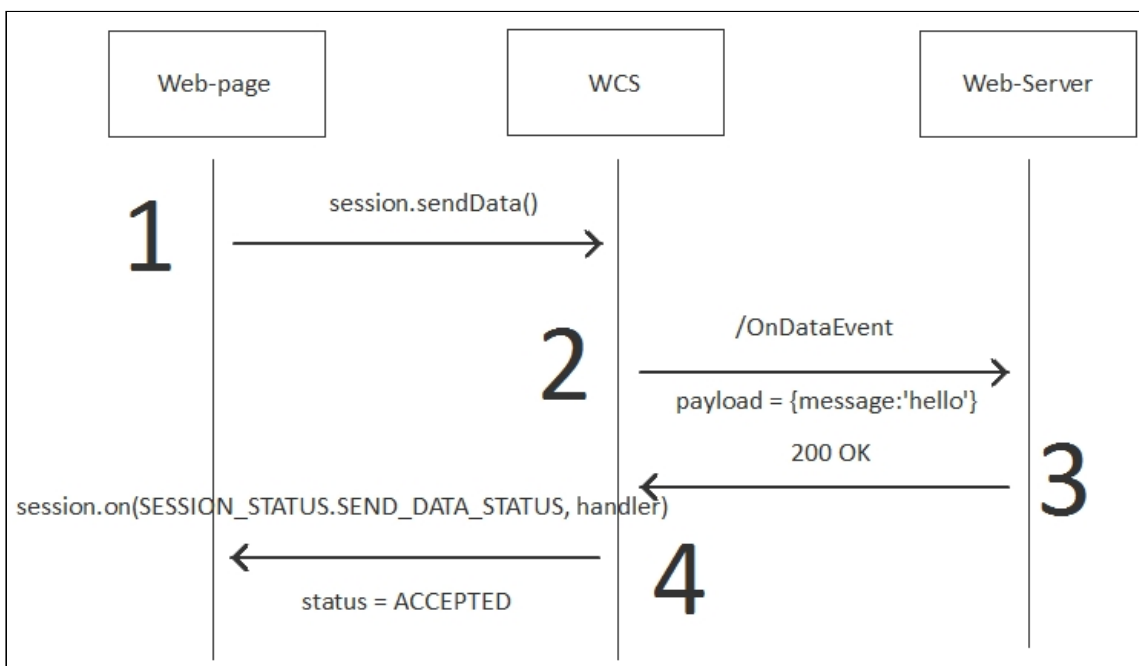


Data exchange - OnDataEvent

A data exchange between a connected clients can be arranged using `Session.sendData()` client function and `OnDataEvent` REST hook. If clients `sessionId` is known, an arbitrary data can be sent to the client using `/rest-api/data/send` REST API query.

Sending data to the server



1. Client sends an arbitrary data object to server using the direct method `Session.sendData()`
2. WCS sends `/OnDataEvent` REST hook to backend server
3. Backend server returns `200 OK`
4. WCS sends `SESSION_STATUS.SEND_DATA_STATUS` with `ACCEPTED` value to the client

Example:

REST hook

```
POST /rest/my_api/OnDataEvent HTTP/1.1
Accept: application/json
Content-Type: application/json;charset=UTF-8
User-Agent: Java/1.8.0_111
```

```
Host: 192.168.1.101
Connection: keep-alive
Content-Length: 218

{
  "nodeId": "Hw47CFMBEchVOpBMDr29IxjudnJ1sj0Y@192.168.1.101",
  "appKey": "defaultApp",
  "sessionId": "/192.168.1.102:55839/192.168.1.101:8443",
  "operationId": "d1999750-fde9-11e6-9f1b-913210792931",
  "payload": {
    "message": "hello"
  }
}
```

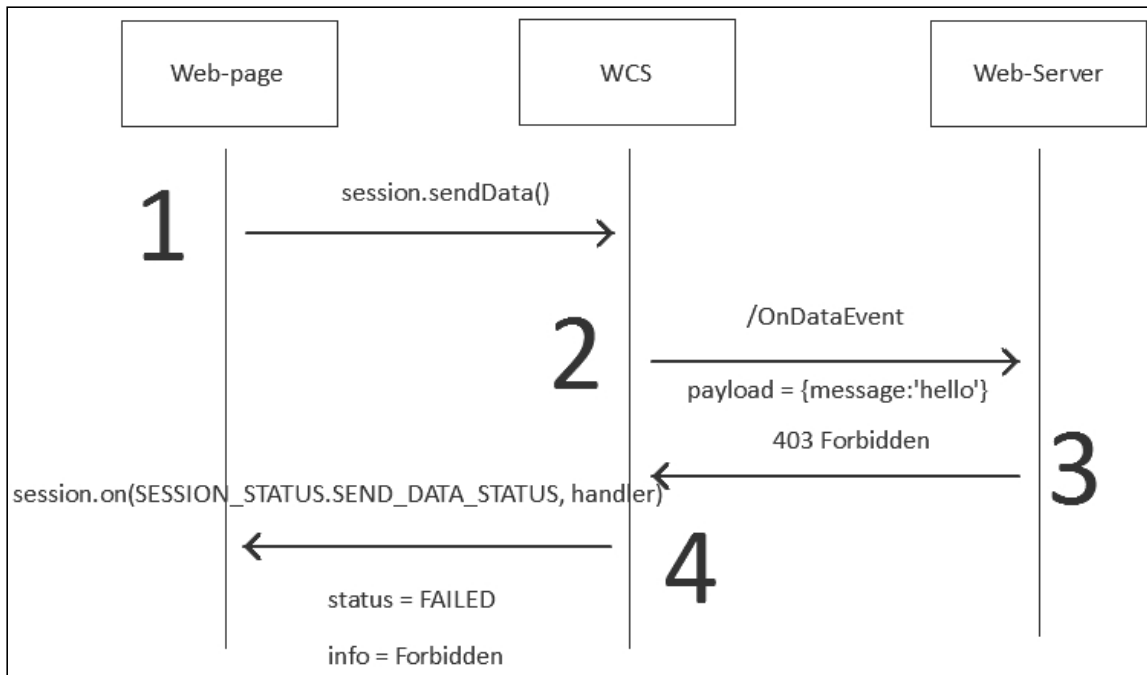
Backend response

```
HTTP/1.1 200 OK
Date: Tue, 28 Feb 2017 19:12:14 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Content-Length: 220
Connection: close
Content-Type: application/json

{
  "nodeId": "Hw47CFMBEchVOpBMDr29IxjudnJ1sj0Y@192.168.1.101",
  "appKey": "defaultApp",
  "sessionId": "/192.168.1.102:55839/192.168.1.101:8443",
  "operationId": "d1999750-fde9-11e6-9f1b-913210792931",
  "payload": {
    "message": "hello"
  }
}
```

Error handling

To make WCS react to the error, set `restOnError: FAIL` in the settings of the `OnDataEvent` method, in `restClientConfig` when establishing the connection.



1. Client sends an arbitrary data object to server using the direct method `Session.sendData()`
2. WCS sends `/OnDataEvent` REST hook to backend server
3. Backend server returns `403 Forbidden`
4. WCS sends `SESSION_STATUS.SEND_DATA_STATUS` with `FAILED` value to the client

Example:

REST hook

```

POST /rest/my_api/OnDataEvent HTTP/1.1
Accept: application/json
Content-Type: application/json;charset=UTF-8
User-Agent: Java/1.8.0_111
Host: 192.168.1.101
Connection: keep-alive
Content-Length: 218

{
  "nodeId": "Hw47CFMBEchVOpBMDr29IxjudnJ1sj0Y@192.168.1.101",
  "appKey": "defaultApp",
  "sessionId": "/192.168.1.102:25789/192.168.1.101:8443",
  "operationId": "d0149310-fdeb-11e6-9b58-9509528e5d66",
  "payload": {
    "message": "hello"
  }
}
  
```

Backend response

```
HTTP/1.1 403 Forbidden
Date: Tue, 28 Feb 2017 19:26:30 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Sending data to connected client

There is a way to send a direct message to a connected client with REST API query `http://host:8081/rest-api/data/send`

To do this, pass the following JSON object

```
{
  "nodeId": "",
  "sessionId": "/192.168.1.102:25789/192.168.1.101:8443",
  "operationId": "",
  "payload": {
    "test": "test"
  }
}
```

Where `sessionId` is connected client session identifier that is received on backend while REST hook `/connect` handling.

In this case, the connected client receives the `custom` object where you can put any data, for example `{"test": "test"}` as shown above.

The client receives the `SESSION_STATUS.APP_DATA` event.

