Controlling REST methods

Switching to your own web server

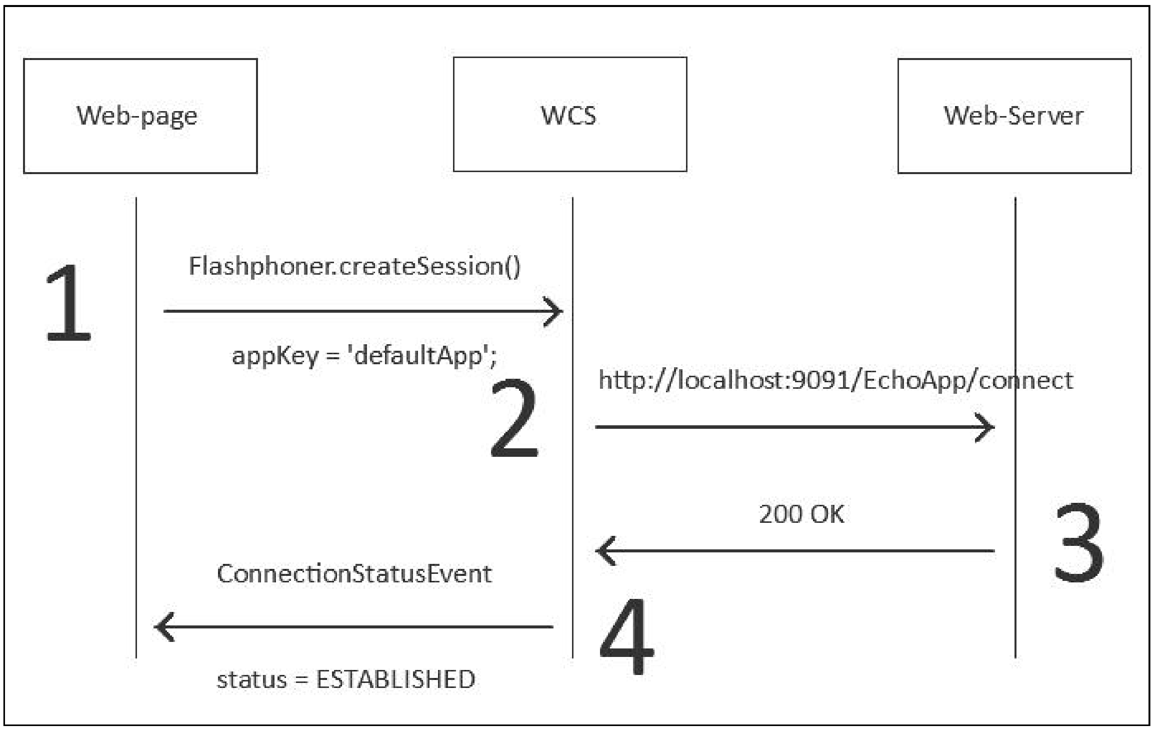
By default, all REST hook queries are sent to the local address: `http://localhost:8081/apps/EchoApp`

The invocations of REST methods look as follows then:

- `http://localhost:8081/apps/EchoApp/connect`
- `http://localhost:8081/apps/EchoApp/playStream`
- `http://localhost:8081/apps/EchoApp/publishStream`

and so on.

This happens because the `appKey: "defaultApp"` parameter is sent when connecting to the WCS server, and the default handler application for this key uses this URL: `http://localhost:8081/apps/EchoApp`



To change these settings, enter to the command line interface of the WCS server core via SSH:

```
ssh -p 2001 admin@localhost
```

Use the password you have specified when installing WCS. The default password is: `admin`

Upon successful authorization, WCS creates a command line:

```
%
```

You can execute the `show apps` command to see the current applications:

```
%show apps
```

Or enter the `help` command to display the list of available commands:

```
%help
```

Then, we create our own REST URL. It is created using the `add app` command. Suppose, we allocated the following address on the web server for REST methods:

- `http://mywebserver.com/rest/connect`
- `http://mywebserver.com/rest/playStream`
- `http://mywebserver.com/rest/publishStream`

and so on.

So we configure this as follows:

```
%add app myApp myAppKey "http://mywebserver.com/rest"
```

This command creates a new URL, `http://mywebserver.com/rest`, and assigns this URL for usage if a client sent the `appKey: "myAppKey"` when establishing a websocket connection.

Therefore, if the newly created `myAppKey` is specified while creating the connection, the control is passed to the assigned URL:

Web SDK client invokation	REST hook URL
`Flashphoner.createSession({appKey:"myAppKey"...});`	`http://mywebserver.com/rest/connect`

By default, server application is created without any REST methods, then methods should be added to application with the following command (`connect` for example):

```
%add app-rest-method MyAppKey connect
```

If backend server will handle all possible REST methods, then all methods should be added with the following command:

```
%add app-rest-method -a MyAppKey
```