

LEARN HOW TO USE CA PPM REST API in 2 Minutes!



WANT TO LEARN MORE ABOUT CA PPM REST API ?

If you are excited about the updates to the REST API in CA PPM V14.4 and would like to explore some of the REST API methods, Annitek will get you started using REST API in 2 minutes.

You will see how easy it is to work with REST even with minimal JavaScript knowledge.

CREATING CA PPM REST API-HTML PORTLETS

Here are the simple steps:

- Login to CA PPM and give yourself the “API – Access” right. You will also need standard rights to any resource or project that you view or update.
- Create a HTML Portlet
- Simply copy and paste scripts Annitek has created for you into the HTML editor within the Portlet and press “Preview”
- Scripts included:
 - POST – AuthToken – Get the authentication Token
 - GET – Get name, code and description of a specific project
 - PATCH – Update description and objective of a specific project
- Notes:
 1. You need to enter your name and password for username/password
 2. You need to know the internal id of the project you need to ‘GET’ the details of, or you intend to modify using the ‘PATCH’ method.

REST API - TOKEN BASED AUTHENTICATION (CA PPM 14.4 on)



```
<script>
var cappm_apiURL = "/ppm/rest/v1";
var username = "xxxxxxxxxxxx"; // Update username
var password = "xxxxxxxxxxxx"; // Update password
var URL = cappm_apiURL + "/auth/login";
var myheader = btoa(username + ":" + password);

var xhr = new XMLHttpRequest();
xhr.open("POST", URL, true);

//Send the proper header information along with the request
xhr.setRequestHeader ("Authorization", "Basic " + myheader);

xhr.onreadystatechange = function() { //Call a function when the state changes.
    if(xhr.readyState == 4 && xhr.status == 200) {

        var myArr = JSON.parse(xhr.responseText);
        var authToken = myArr.authToken;
        alert ("authToken: " + authToken );
    }
}

xhr.send();
</script>
```

Implementation Notes:

Resource for login. Need to pass a user name and password as basic authentication header parameters

Benefit: Once you have this token all the remaining calls can use it. Call the Delete Method to logout. You will need to pass authToken as a parameter for Delete Method.

REST API – GET THE DETAILS OF A SPECIFIC PROJECT

Implementation Notes:
Returns a list of projects.

```
<script>
var cappm_apiURL = "/ppm/rest/v1";
var username = "xxxxxxxxxxxxx"; // Update username
var password = "xxxxxxxxxxxxx"; // Update password
// Update project internal id
var URL = cappm_apiURL + "/projects/5001085?fields=name,code,description";
var myheader = btoa(username + ":" + password);

var xhr = new XMLHttpRequest();
xhr.open("GET", URL, false);
xhr.setRequestHeader("Authorization", "Basic " + myheader) ;

xhr.onreadystatechange = function() { //Call a function when the state changes.
    if(xhr.readyState == 4 && xhr.status == 200) {

        var myArr = JSON.parse(xhr.responseText);
        var name = myArr.name;
        var code = myArr.code;
        var description = myArr.description;
        alert("Project Name: " + name + ", Code: " + code + ", Description: " + description );
        alert(JSON.stringify(JSON.parse(xhr.responseText), null, ' ' ));
    }
}

xhr.send();
</script>
```

REST API – PATCH (MODIFY) THE DETAILS OF A SPECIFIC PROJECT

```
<script>
var cappm_apiURL = "/ppm/rest/v1";
var username = "xxxxxxxxxxxx"; // Update username
var password = "xxxxxxxxxxxx"; // Update password
// Update project internal id
var URL = cappm_apiURL + "/projects/5005000";
var myheader = btoa(username + ":" + password);
var project_data = {

    "description": "Description updated via REST API",
    "objective": "Objective updated via REST API"

};

var xhr = new XMLHttpRequest();
xhr.open("PATCH", URL, true);
xhr.setRequestHeader ("Authorization", "Basic " + myheader) ;
xhr.setRequestHeader ("Content-type", "application/json") ;

xhr.onreadystatechange = function() { //Call a function when the state changes.
    if(xhr.readyState == 4 && xhr.status == 200) {

        alert("Project data modified");
    }
}
xhr.send(JSON.stringify(project_data));
</script>
```

Implementation Notes:

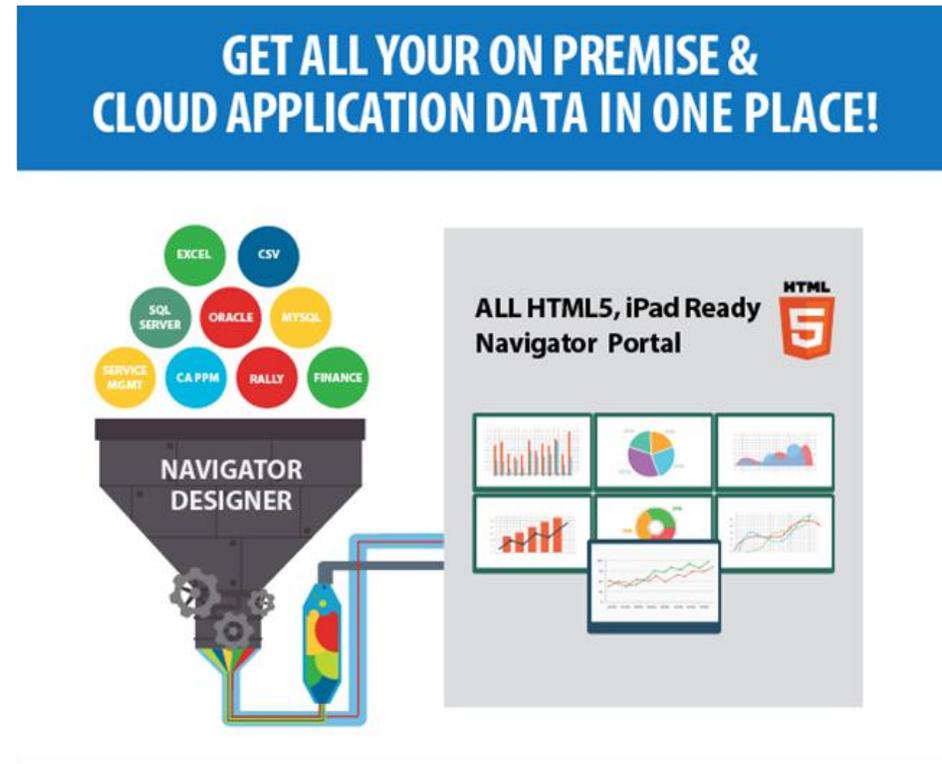
Performs an update operation on the resource {projectsInternalId}.

Tip: You can also use PUT like this to completely update project object.

Notes:

- Running these script from desktop or another application will give you CORS related errors.
 - A resource makes a cross-origin HTTP request when it requests a resource from a different domain than the one which the first resource itself serves. For security reasons, browsers restrict cross-origin HTTP requests initiated from within scripts. Your server will not have CORS enabled, hence the error.
- Keeping brevity of presentation in mind, not every detail has been commented nor have we tried to beautify the scripts. Annitek leaves that to you.

About Annitek



For more information about how Annitek can assist you with REST API integration needs, email: info@annitek.com