5.1 New study Technology Innovation

**1. SuccessFactor OData APIs(V2) Access From POSTMAN**

**2. PREACT** (Fast 3kB alternative to React with the same modern API)

**SuccessFactor OData APIs(V2) Access From POSTMAN**

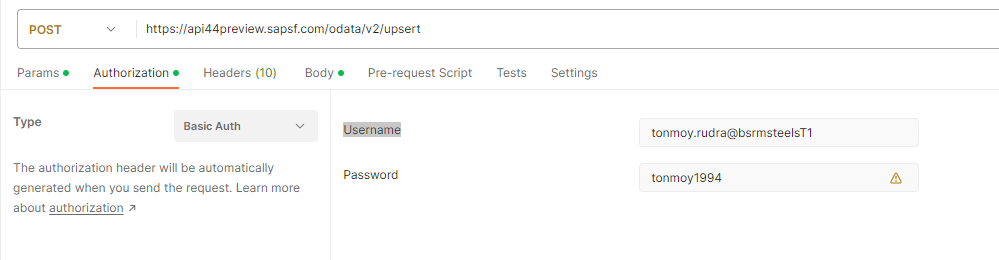
The Open Data Protocol (OData) is a standardized protocol for creating and consuming data APIs.OData builds on core protocols like HTTP, and commonly accepted methodologies like REST. The result is a uniform way to expose full-featured data APIs. OData provides both a standard for how to represent your data and a metadata method to describe the structure of your data, and the operations available in your API. This document focuses on OData version 2.0. [Details](https://help.sap.com/docs/SAP_SUCCESSFACTORS_PLATFORM/d599f15995d348a1b45ba5603e2aba9b/03e1fc3791684367a6a76a614a2916de.html)

# **SAP SuccessFactors API Servers :**

| **Data Center** | **Environment** | **Location** | **API Server** |
| --- | --- | --- | --- |
| DC44 | Preview | Singapore | https://api44preview.sapsf.com/ |
| DC44 | Production | Singapore | https://api44.sapsf.com/ |

**Step 1:** First You need to authenticate with Basic Authentication. So You need to enter

Username = [userName@companyID], Password: [yourPassword] on Basic Auth Type.

**Example** : 

**Step 2:** Now I want to get email address for a user. So how can I do this?

HTTP Method: GET

URL: https://api44preview.sapsf.com/odata/v2/PerEmail?$filter=emailType eq

'22151' and personIdExternal eq '100628' and isPrimary eq 'true'&$select=emailAddress&$format=JSON

Response:

{

    "d": {

        "results": [

            {

                "\_\_metadata": {

                    "uri": "https://api44preview.sapsf.com/odata/v2/PerEmail(emailType='22151',personIdExternal='100628')",

                    "type": "SFOData.PerEmail"

                },

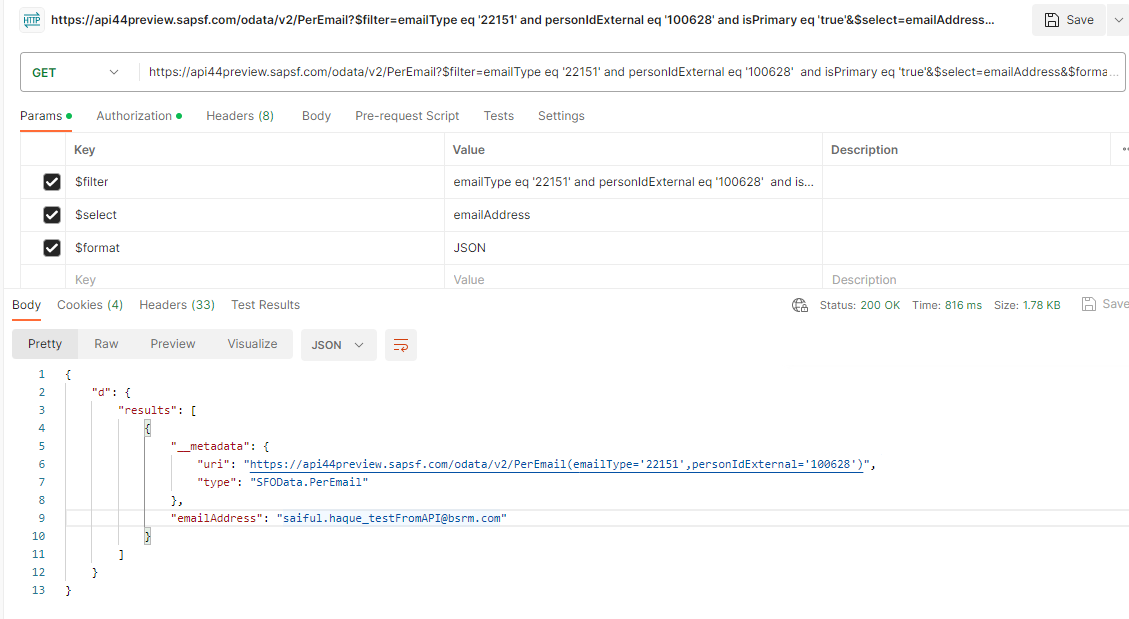
                "emailAddress": "saiful.haque@bsrm.com"

            }

        ]

    }

}

Example: 

**Step 3:** Now I update email address for this user.

Method: POST

URL: <https://api44preview.sapsf.com/odata/v2/upsert>

Req BODY: {

    "\_\_metadata": {

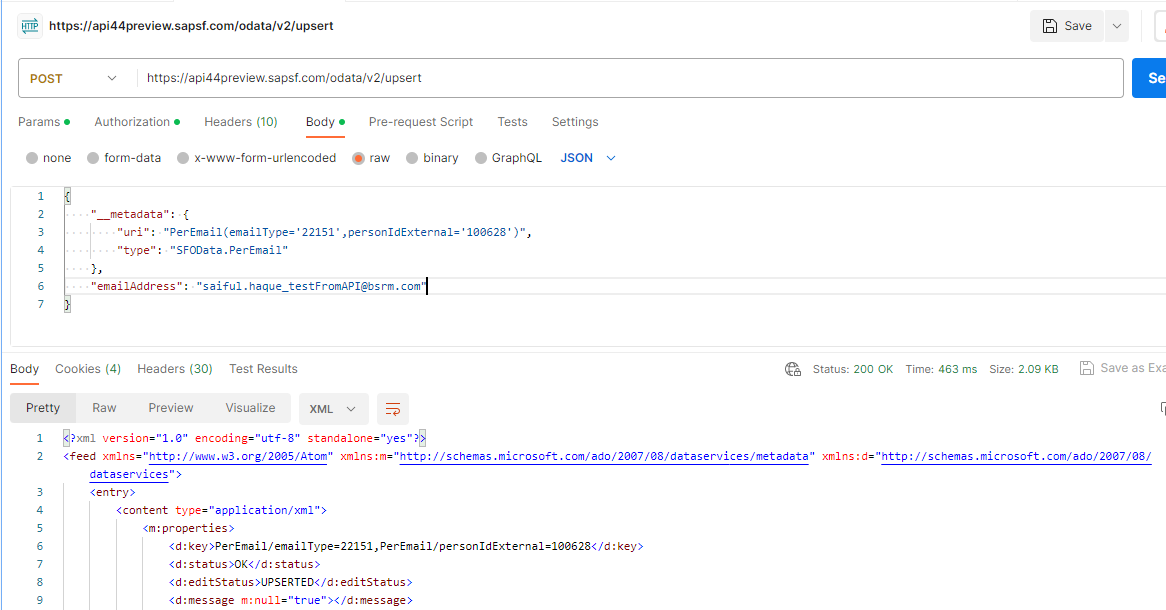
        "uri": "PerEmail(emailType='22151',personIdExternal='100628')",

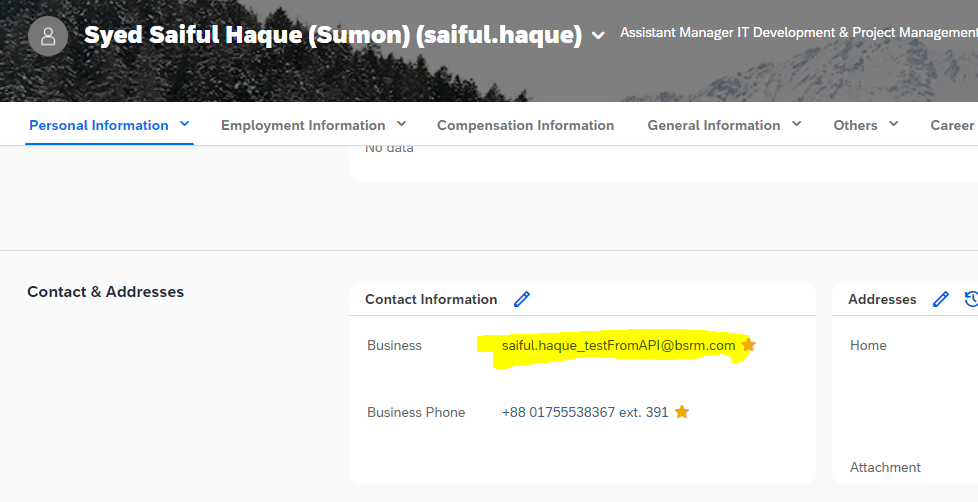
        "type": "SFOData.PerEmail"

    },

    "emailAddress": "saiful.haque\_testFromAPI@bsrm.com"

}

Example: 

And email is updated. 

Ref Documentation: <https://help.sap.com/docs/SAP_SUCCESSFACTORS_PLATFORM/d599f15995d348a1b45ba5603e2aba9b/03e1fc3791684367a6a76a614a2916de.html>

**PREACT**

(Fast 3kB alternative to React with the same modern API)

**What is Preact?**

Preact is a fast, lightweight alternative to React. It’s smaller and more straightforward than React, so it’s easier to learn and use. Preact is also more efficient and faster and uses less memory. If you’re looking for a simple, lightweight option for your web app or website, Preact is an excellent choice. It’s easy to learn and use and more efficient than React, so it’ll run faster and use less memory.

**Feature of Preact:**

**Closer to the DOM:**

Preact provides the thinnest possible Virtual DOM abstraction on top of the DOM. It builds on stable platform features, registers real event handlers and plays nicely with other libraries. Preact can be used directly in the browser without any transpilation steps.

**Small Size:**

Most UI frameworks are large enough to be the majority of an app's JavaScript size. Preact is different: it's small enough that your code is the largest part of your application.

That means less JavaScript to download, parse and execute - leaving more time for your code, so you can build an experience you define without fighting to keep a framework under control.

**Big Performance:**

Preact is fast, and not just because of its size. It's one of the fastest Virtual DOM libraries out there, thanks to a simple and predictable diff implementation. We automatically batch updates and tune Preact to the extreme when it comes to performance. We work closely with browser engineers to get the maximum performance possible out of Preact.

**Portable & Embeddable:**

Preact's tiny footprint means you can take the powerful Virtual DOM Component paradigm to new places it couldn't otherwise go. Use Preact to build parts of an app without complex integration. Embed Preact into a widget and apply the same tools and techniques that you would to build a full app.

**Instantly Productive:**

Lightweight is a lot more fun when you don't have to sacrifice productivity to get there. Preact gets you productive right away. It even has a few bonus features:

* props, state and context are passed to render()
* Use standard HTML attributes like class and for.

**A Quick Comparison of Preact and React**

React is a modern library created by Facebook. It has a large community and is used by some of the biggest companies in the world, including Facebook, Netflix, and Airbnb. React is a complete solution for creating user interfaces. It has many features and can be used with different programming paradigms.

Preact is a smaller alternative to React that includes only the essentials for creating user interfaces. It’s much lighter-weight than React and has a smaller footprint. Preact is also faster than React, which can be important for performance-sensitive applications.

So, which library should you choose? That depends on your specific needs. If you need a complete solution with a wide range of features, React is probably the better choice. If you’re looking for a lighter-weight alternative that’s still feature-rich and performant, Preact might be the right choice for you.

**Pros and Cons of Preact**

Preact is a fast, lightweight alternative to React that offers the same benefits. However, you should be aware of some potential drawbacks to using Preact before making the switch.

**PROS:**

-Offers the same benefits as React

-Fast and lightweight

-Simple and easy to use

It-Great for small projects or prototyping

**CONS:**

-May not be suitable for large projects

-Lack of community support

**Conclusion**

Preact and React are both great options for front-end development, but they have different strengths. Preact is smaller and faster, while React has more features. If you’re looking for a simple, lightweight option, Preact is a good choice. If you need more parts or want to use the latest technology, React is a better option.