**CitiBanamex API Factory Welcome KIT**

[BGV 2](#_Toc483468696)

[Onboarding 2](#_Toc483468697)

[Anti-Bribery compliance 2](#_Toc483468698)

[Trainings 3](#_Toc483468699)

[Wipro IS training (Mandatory) 3](#_Toc483468700)

[CitiBanamex Overview 4](#_Toc483468701)

[Agile Training Document 4](#_Toc483468702)

[API Development Project Plan Template 4](#_Toc483468703)

[API Developer Guide V3 4](#_Toc483468704)

[API Best Practices 4](#_Toc483468705)

[API Software Installation Laptop / Desktop 5](#_Toc483468706)

[API Development 5](#_Toc483468707)

[Build tool 5](#_Toc483468708)

[Swagger API 5](#_Toc483468709)

[CI/CD 5](#_Toc483468710)

[Version Control 5](#_Toc483468711)

[SSH Connection tool 6](#_Toc483468712)

[Unit Rest API testing 6](#_Toc483468713)

[AML (Application Lifecycle management) 6](#_Toc483468714)

[Automated testing of Rest API 6](#_Toc483468715)

[Pushing Code to Bluemix 7](#_Toc483468716)

[API Testing 8](#_Toc483468717)

[API Study Material 9](#_Toc483468718)

[What is an API ? 9](#_Toc483468719)

[How does an API differ from a MicroService ? 10](#_Toc483468720)

[How does Api-fication impact the enterprise ? 12](#_Toc483468721)

[OK So what!!! Where do I Start? 13](#_Toc483468722)

[Java Classroom Trainings. 16](#_Toc483468723)

[Java Online Sessions. 17](#_Toc483468724)

# **BGV**

Please fill the details on iverify.

<https://iverify.wipro.com/iverify/>

# **Onboarding**

Please send the below list of documents to [phaneendrakumar.tadikonda@wipro.com](mailto:phaneendrakumar.tadikonda@wipro.com)

1. CitiBanamex Onboarding template with personal info.
2. Identity Proof – India Pan Card / Passport or Mexico Identity proof.
3. Wipro Resume



## **Anti-Bribery compliance**

This training should be completed to initiate the VISA processing.

Please follow this link:

<https://newitms.wipro.com> >> MANDATORY LEARNING >> Anti-Corruption to complete the assessment

# **Trainings**

## **Wipro IS training (Mandatory)**

Please complete the **Wipro IS training (Information Security Risk & Compliance e-Learning Program)** course and share the certificate or course completion screen shot.

|  |  |
| --- | --- |
| **How to access the e-learning curriculum?** | **Whom to contact in case of issues?** |
| **• If you are accessing from Wipro network:** [**https://newitms.wipro.com**](https://newitms.wipro.com/)**>>Learning Initiatives>>IRMC >>  Information Security Risk & Compliance E-learning Program**  **• If you are accessing from non-Wipro network:** [**https://Gateway.wipro.com/itms**](https://gateway.wipro.com/itms)**>> ITMS (e-Learning) >>  Learning Initiatives>>IRMC >>  Information Security Risk & Compliance E-learning Program Or** [**https://betagateway.wipro.com/itms**](https://betagateway.wipro.com/itms)**>> ITMS (e-Learning) >> Learning Initiatives>>IRMC >>  Information Security Risk & Compliance E-learning Program** | **• Please log a ticket in helpline:** [**https://helpline.wipro.com**](https://helpline.wipro.com/) **You could also call the 24\*7 Help Desk. Toll Free: India :- 1800 3000 6666  ITFS UK :- 8081683862  ITFS US : -18558546384  VOIP :- 829-6666** |
| **Prerequisites for accessing ITMS   The path to Download the Pre-requisites are:** [**http://helpline.wipro.com**](http://helpline.wipro.com/) **>> Self Service >> Solutions >> ITMS Ready Reckoner From ITMS >>** [**http://itms.wipro.com/access.aspx**](http://itms.wipro.com/access.aspx) **>> System Readiness Check (Right Hand Panel)** | |

# **CitiBanamex Overview**



# **Agile Training Document**



# **API Development Project Plan Template**



# **API Developer Guide V3**



# **API Best Practices**



**API Software Installation Laptop / Desktop**

**\*\*\*\*\* All software should be 64 bit support**

## **API Development**

Spring Tool Suite - 3.8.3

JDK - 8u121

Spring & SpringBoot Frameworks

Apache Tomcat - 8

Notepad++ - 7.3.1 (not sure if CATE certified)

## **Build tool**

Apache Maven - 3.3.9

## **Swagger API**

Swagger Core - 1.5.10

Swagger Editor - 2.10.4

Swagger UI - 2.2.10

Swagger Codegen - 2.2.1

Node JS - V6

Jackson - 2.4.5

## **CI/CD**

Jenkins - 2.32.2

## **Version Control**

Apache Subversion - 1.9

TortoiseSVN - 1.9.5

## **SSH Connection tool**

Tectia SSH - 6.4

## **Unit Rest API testing**

SOAP UI - 5.3

## **AML (Application Lifecycle management)**

RTC (IBM)

## **Automated testing of Rest API**

HPE UFT - (Integrated with HP QC)

# **Pushing Code to Bluemix**

Team,

Here are some options on how to push Java apps to Bluemix. Please cascade.

1. There is an Eclipse plugin for Bluemix that can be downloaded from the link below and installed on your IDE. The plugin will need your Bluemix id and password and push maven projects to Bluemix creating a CF app.

<https://www.ibm.com/cloud-computing/bluemix/eclipse>

1. Download the CF CLI installer and install it on your laptops. Then follow the series of commands to push code to Bluemix. All your source Java files and Maven build files should be in one folder. Then execute cf push from this folder (syntax below). PCF will recognize .pom file and invoke the Java buildpack to build your code and deploy it on Liberty container and start the app. You can also refer to sample projects as there will be initial hiccups to deploy. If you have already created a WAR file, you can deploy on Bluemix using the following CLI commands:

cf api <https://api.ng.bluemix.net/>

cf login

Deploy .war to Bluemix by running this command:

cf push your app name -p myapp.war

Here  is a link that shows how to deploy a sample project using CLI

<https://www.ibm.com/developerworks/cloud/library/cl-bluemix-fundamentals-create-and-deploy-a-java-app-to-the-cloud/>

1. Create a CF app on Bluemix and select Liberty for Java. In the app overview page, click ADD GIT and create a repos for your app. From command line push your Java sources to GIt repos. E.g

Clone the remote repos on your laptop as

git clone [https://hub.jazz.net/git/jazzhubdemouser/<your\_app](https://hub.jazz.net/git/jazzhubdemouser/%3cyour_app)>

Then use the following commands from command line to push your code to git repos

$ cd /project\_directory

$ git stage .

$ git commit -m "your\_app"

$ git push

Regards,

Sujoy

# **API Testing**

1- SoapUI Tool (Opensource)

2- Groovy Scripting, JAVA, JavaScript Knowledge to customize the output as per project requirement.

3- JXL.JAR file (available at below link)

Link: <http://www.java2s.com/Code/Jar/j/Downloadjxljar.htm>

4- Java Version 6 or above

# **API Study Material**

## **What is an API ?**

An API is an interface that allow applications/services to communicate with each other.

The Application Programming Interface is best thought of as a contract provided by one piece of computer software to another.

Many things have APIs nowadays. Your operating system of choice has a really big list of APIs. Your smartphone has a bunch of APIs too. Websites can have APIs as well. It's important to note that pieces of software can interact with an API.

For example:

An application finds the current weather in London by sending a message to the weather.com API (in a structured format like JSON). The weather.com API then replies with a structured response. With an API, the exact structure of request and response is documented upfront by weather.com and is likely to remain constant, regardless of whether the website changes its look and feel for human visitors.

Basically if you want to access any services outside of your application, you need an API to access its services.

In layman's terms, an API is an agreement between two people stating: "If you give me this instruction, I will perform this action, or return this information

Please check the below URL's for API

<https://www.youtube.com/watch?v=s7wmiS2mSXY>

<https://www.youtube.com/watch?v=4JjN54aaF74>

<https://www.youtube.com/watch?v=QSUnBPv4iQ0>

<https://www.youtube.com/watch?v=TbVtliFXOOY>

## **How does an API differ from a MicroService ?**

MicroService is an independent component of an application that provides business services within the scope of the application.

Micro-services are independently-created and independently-maintained components that communicate with one another through contractually agreed-upon interfaces — APIs, short for “application program interfaces.”

Microservices typically provide fine-grained APIs, which means that clients need to interact with multiple services.

For example let’s imagine that you are building an e-commerce application that takes orders from customers, verifies inventory and available credit, and ships them.

The application consists of several components including the StoreFrontUI, which implements the user interface, along with some backend services for checking credit, maintaining inventory and shipping orders. The application is deployed as a set of services.

A client who asks for the details of a product needs to fetch data from numerous services like

Product Info Service - basic information about the product such as title, author

Pricing Service - product price

Order service - purchase history for product

Inventory service - product availability

Review service - customer reviews …

Consequently, the code that displays the product details needs to fetch information from all of these services.

Each microservice is relatively small.

Microservice is easier for a developer to understand.

Each Microservice can be deployed independently of other services - easier to deploy new versions of services frequently

Easier to scale development. It enables you to organize the development effort around multiple teams. Each team is responsible a single service.

Each team can develop, deploy and scale their service independently of all of the other teams

Improved fault isolation. For example, if there is a memory leak in one service then only that service will be affected. The other services will continue to handle requests.

API's are interfaces that allow microservices to communicate with each other or other applications.

APIs don’t just allow individual programs and services to communicate, they encourage it to do so from business perspective.

<http://www.ibm.com/developerworks/websphere/library/techarticles/1601_clark-trs/1601_clark.html>

<http://thenextweb.com/insider/2015/08/26/apis-and-micro-services-how-everything-you-do-online-is-changing/>

<http://microservices.io/patterns/microservices.html>

## **How does Api-fication impact the enterprise ?**

Business needs are driving enterprises to open their data and applications more to partners, developers, mobile apps, and cloud services.

APIs provide a standardized way to open up information assets across the Web, mobile devices, and the cloud. APIs open up new opportunities for executives to evaluate business information resources to build value for the enterprise by creating services that others can consume

For example Salesforce.com generates 50% of its revenues through APIs. Expedia.com generates 90%. eBay generates 60%

Please check below urls for more impact of apis on the enterprise business

<http://blogs.ca.com/2014/05/15/the-increasing-impact-of-apis-on-the-digital-marketplace/>

<https://www.infoq.com/articles/web-apis-business-perspective>

<https://medium.com/@medjawii/5-ways-an-api-is-more-than-an-api-bddcdb0517ca#.mtd33g4vj>

<https://www.ca.com/us/rewrite/articles/application-economy/slideshare--5-technical-aspects-of-apis-that-impact-your-busines.register.html>

<https://pages.apigee.com/rs/351-WXY-166/images/apigee-state-of-APIs-report-2016-03.pdf>

## **OK So what!!! Where do I Start?**

Understanding the opportunities to capitalize on APIs is crucial, and many companies are turning to their marketing departments for direction on API strategies. To their credit, marketing professionals are often forward-looking and trend-driven, and many have been monitoring API advances and see the business opportunities potentially enabled by API adoption. They are well positioned to evaluate available data within business applications to determine what information the organization is willing to make available in a safe and secure manner to create value for the enterprise.

<https://www.ca.com/content/dam/ca/us/files/ebook/api-strategy-and-architecture-a-coordinated-approach.pdf>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Microservice Pros and Cons**

Microservices are not a silver bullet, and by implementing them you will expose communication, teamwork, and other problems that may have been previously implicit but are now forced out into the open. But API Gateways in Microservices can greatly reduce build and qa time and effort.

One common issue involves sharing schema/validation logic across services.  What A requires in order to consider some data valid doesn’t always apply to B, if B has different needs.  The best recommendation is to apply versioning and distribute schema in shared libraries.  Changes to libraries then become discussions between teams.  Also, with strong versioning comes dependencies, which can cause more overhead.  The best practice to overcome this is planning around backwards compatibility, and accepting regression tests from external services/teams.  These prompt you to have a conversation before you disrupt someone else’s business process, not after.

As with anything else, whether or not microservice architecture is right for you depends on your requirements, because they all have their pros and cons.  Here’s a quick rundown of some of the good and bad:

**Pros**

Microservice architecture gives developers the freedom to independently develop and deploy services

A microservice can be developed by a fairly small team

Code for different services can be written in different languages (though many practitioners discourage it)

Easy integration and automatic deployment (using open-source continuous integration tools such as Jenkins, Hudson, etc.)

Easy to understand and modify for developers, thus can help a new team member become productive quickly

The developers can make use of the latest technologies

The code is organized around business capabilities

Starts the web container more quickly, so the deployment is also faster

When change is required in a certain part of the application, only the related service can be modified and redeployed—no need to modify and redeploy the entire application

Better fault isolation: if one microservice fails, the other will continue to work (although one problematic area of a monolith application can jeopardize the entire system)

Easy to scale and integrate with third-party services

No long-term commitment to technology stack

**Cons**

Due to distributed deployment, testing can become complicated and tedious

Increasing number of services can result in information barriers

The architecture brings additional complexity as the developers have to mitigate fault tolerance, network latency, and deal with a variety of message formats as well as load balancing

Being a distributed system, it can result in duplication of effort

When number of services increases, integration and managing whole products can become complicated

In addition to several complexities of monolithic architecture, the developers have to deal with the additional complexity of a distributed system

Developers have to put additional effort into implementing the mechanism of communication between the services

Handling use cases that span more than one service without using distributed transactions is not only tough but also requires communication and cooperation between different teams

The architecture usually results in increased memory consumption

Partitioning the application into microservices is very much an art

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Pros and cons of Node JS

-------------------------

<http://voidcanvas.com/describing-node-js/>

<http://blog.carbonfive.com/2013/10/27/the-javascript-event-loop-explained/>

<http://www.hostreview.com/blog/160311-the-pros-and-cons-of-using-nodejs>

Microservice

--------------

<https://dzone.com/articles/scalable-cloud-computing-with-microservices>

<http://www.developer.com/open/building-microservices-with-open-source-technologies.html>

<https://techbeacon.com/8-best-open-source-tools-building-microservice-apps>

API

------

<https://dzone.com/articles/using-jax-rs-with-spring-boot-instead-of-mvc>

API - governance

-----------------

<https://apigee.com/about/cp/api-governance>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**What is an api?**

<https://www.youtube.com/watch?v=s7wmiS2mSXY>

<https://www.youtube.com/watch?v=6STSHbdXQWI>

<https://www.youtube.com/watch?v=FknvOGcLHmc>

<https://www.youtube.com/watch?v=B9vPoCOP7oY>

**Api gateway&microservices**

<https://www.youtube.com/watch?v=z_aCpGqt3U8>

**What is microservices?**

<https://www.youtube.com/watch?v=PY9xSykods4>

**Api-fication and enterprise**

<http://www.slideshare.net/SOA_Software/the-business-value-for-internal-apis-in-the-enterprise>

<https://www.mashery.com/blog/api-management-20-empowering-enterprise-third-industrial-revolution>

<http://www.ibm.com/developerworks/websphere/library/techarticles/1601_clark-trs/1601_clark.html>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## **Java Classroom Trainings.**

Please refer an email from the below Wipro ID and nominate for the necessary trainings at your work location.

Competency GroupFS [competency.fsbu@wipro.com](mailto:competency.fsbu@wipro.com)

## **Java Online Sessions.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Greetings from FS Competency Group!!**  FS Competency Group is pleased to bring Core Java and Advanced Java online sessions for continuous enablement of employees across Geo.These training sessions are focused to reskill/ Upskill employees in Java programming/ provide learning opportunity to Non-Java programmers thereby providing them opportunity to meet business demands in Java.    **METTL assessment** for self-evaluation will be enabled on request post successful completion of training to leverage the knowledge gained from the training.  **Online trainings**   |  |  |  | | --- | --- | --- | | **Skill** | **Learning path** | **Topgear link** | | **Restful Web Services** | <https://newitms.wipro.com> -> Technology ->WebServices  **Curriculum code: 2509** |  | | **Service Oriented Architecture** | <https://newitms.wipro.com> -> Technology ->WebServices  **Curriculum code: 2654** |  | | **SOA-WebServices 1.1** | <https://newitms.wipro.com> -> Technology ->WebServices  **Curriculum code: 377** |  | | **JQuery** | <https://newitms.wipro.com>->Learning Initiatives -> Front End / User Interface Frontend ->  **Curriculum code: 2699** | <https://topgear-app.wipro.com/spring-community/content/spring-l1-training> | | **Bootstrap JS** | <https://newitms.wipro.com>->Learning Initiatives > Front End / User Interface Frontend ->  **Curriculum code: 2729** | <https://topgear-app.wipro.com/open-source-java/content/core-java-l1-training-1> | | **AngularJS** | <https://newitms.wipro.com> -> Learning Initiatives > Technology > JAVA > Technology-Java: AngularJS (Day-1)  **Curriculum code: 2309** | <https://topgear-app.wipro.com/angular-js/content/angular-js-l1-training-0> | | **NodeJS** | <https://newitms.wipro.com> -> Learning Initiatives > WebEx Recording>  Intro to NodeJs & ExpressJS  **Curriculum code: 22536** | <https://topgear-app.wipro.com/nodejs/content/nodejs-l1training-0> | | **Spring** | <https://newitms.wipro.com>->Learning Initiatives > Technology > Spring Framework >  Technology: Spring Framework  **Curriculum code: 2165** | <https://topgear-app.wipro.com/spring-community/content/spring-l1-training> | | **Core Java** | <https://newitms.wipro.com>->Learning Initiatives > Technology > JAVA  >  WA: Java Programming  **Curriculum code: 2248** | <https://topgear-app.wipro.com/open-source-java/content/core-java-l1-training-1> | |