# Hands-on Lab:
# Automated Staging with Git and Jenkins

## Objectives

In this hands-on lab, you will create a local repository by exporting all assets from the **API Gateway Dev** environment with the **Asset Build Environment**. You will commit the local repository to **Git**. Then you will configure a **Jenkins** job which will be triggered by changes in the **Git** repository. The **Jenkins** job will automatically create a repository out of the local repository and deploy all assets to the **API Gateway Prod** environment using the **webMethods Deployer** command-line interface.

## Steps

1. Open Windows Services UI to double check that the following services, needed for API Gateway and the native services, are up and running. If service is not running, start the service.
    a) **Software AG Dev Stage Integration Server 10.5**
    b) **Software AG Prod Stage Integration Server 10.5**
    c) **Software AG Internal Integration Server 10.5**
2. Open the **Windows Explorer** and navigate to **C:\Training\E456C04-75E\setup**.
3. Right-click file **Restore_Initial_Backup_PROD.bat** and select **Run as administrator**. This will restore the initial (empty) backup of the **API Gateway Prod Stage** database created in hands-on lab **"Prepare Bookstore API"**.
4. Navigate to folder **C:\apigatewayRepoForDeployer** and remove its entire content.
5. Open the Windows Explorer and navigate to folder **C:\SoftwareAG\common\AssetBuildEnvironment\master_build**. Right-click file **build.properties** and select **Edit with Notepad++**. Go to line 63 and ensure that the value of **enable.build.APIGateway** is set to **true**. If necessary, change the value and save the file.
6. Open a Windows **Command Prompt** (cmd) as administrator and navigate to the **Asset Build Environment** bin folder:

    ```
    cd \SoftwareAG\common\AssetBuildEnvironment\bin
    ```

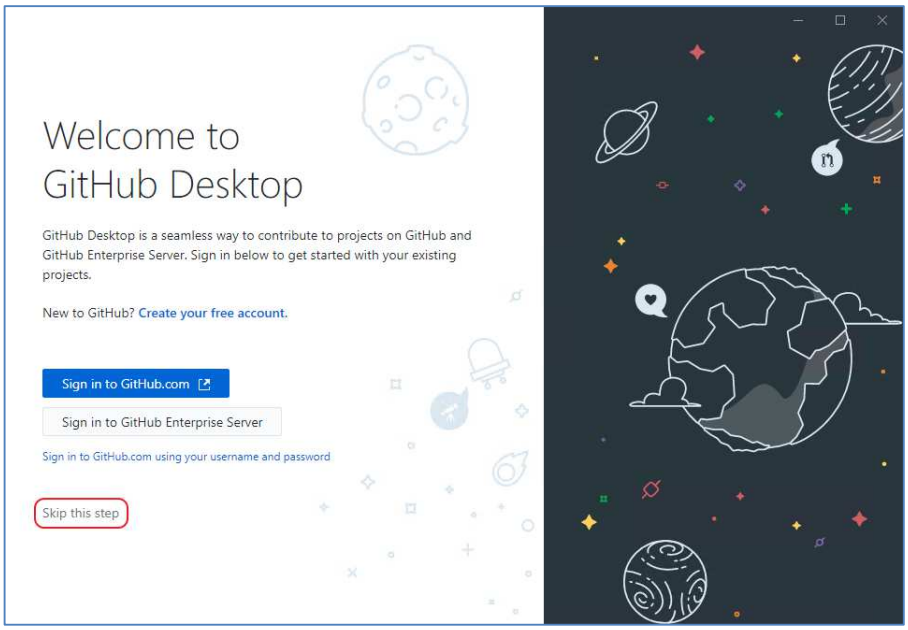
7. Run the **Asset Build Environment** build script to create a local repository including all assets from the **API Gateway Dev** environment:

    ```
    build.bat -Dbuild.output.dir=C:\apigatewayLocalRepo\
    -Dapigateway.is.url=http://localhost:5555
    -Dapigateway.is.username=Administrator
    -Dapigateway.is.password=manage
    -Dapigateway.buildLocalRepoOnly=true
    ```

> *Note*: You can copy and paste the command from the file **build_and_deployment.txt** in **C:\Training\E456C04-75E\Lab8**.

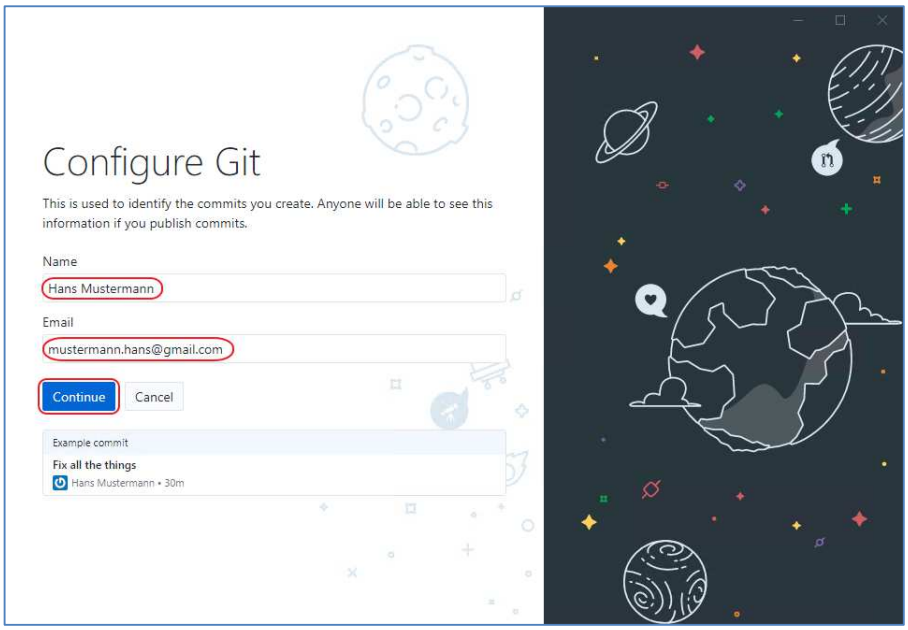
```
C:\SoftwareAG\common\AssetBuildEnvironment\bin>build.bat -Dbuild.output.dir=C:\apigatewayLocalRepo\ -Dapigateway.is.url=
http://localhost:5555 -Dapigateway.is.username=Administrator -Dapigateway.is.password=manage -Dapigateway.buildLocalRepo
Only=true
Buildfile: C:\SoftwareAG\common\AssetBuildEnvironment\master_build\build.xml

check-build.prop-loc:

initLog:

build:
     [echo] Apache Ant(TM) version 1.10.5 compiled on July 10 2018

createNewBuildNumber:

createBuildNumber:
[propertyfile] Updating property file: C:\SoftwareAG\common\AssetBuildEnvironment\master_build\build.number

createOutputDir:
     [echo] Creating directory C:\apigatewayLocalRepo\
    [mkdir] Created dir: C:\apigatewayLocalRepo
     [echo] Started APIGateway Build.............

buildAPIGateway:

build:

validateParameters:

createLocalRepository:

downloadFromSourceAPIGateway:
     [java] OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128M; support was removed in 8.0
     [echo] APIGateway Build complete...............

check-build.prop-loc:

createIndex:
[createIndex] RepoIndexer is processing composite Assets for API Gateway from composite file C:\apigatewayLocalRepo\APIG
atewayAssets.acdl

BUILD SUCCESSFUL
Total time: 3 seconds
```

8. You might want to inspect the repository created in **C:\apigatewayLocalRepo**.
9. Start the **GitHub Desktop Git** client.

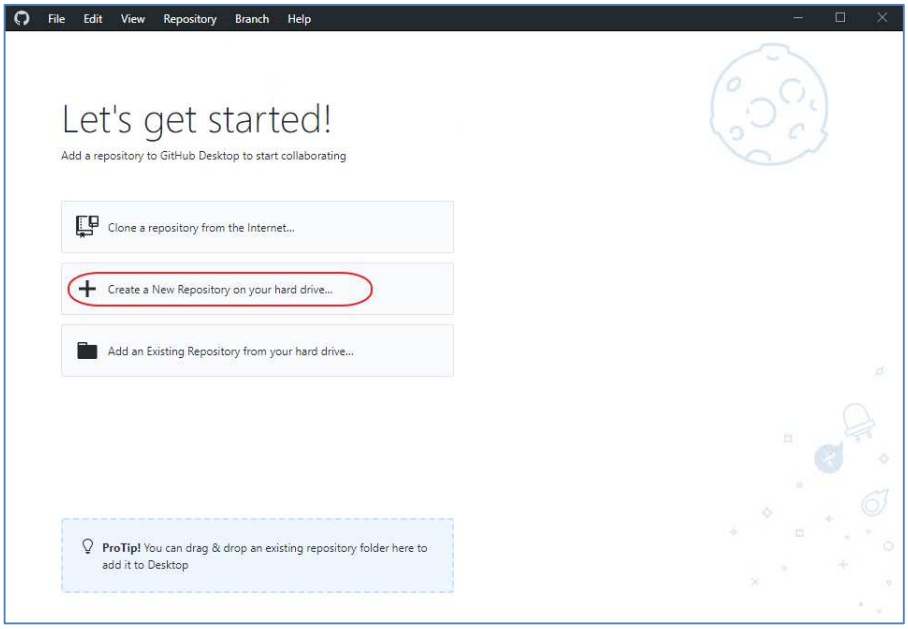



Click on **Skip this step**.

10. Enter your name and your e-mail address.




Click **Continue**. Click **Finish**.

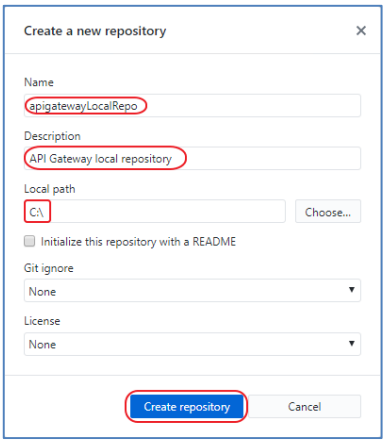


11. Click on **Create a New Repository on your hard drive…** and provide the following properties:

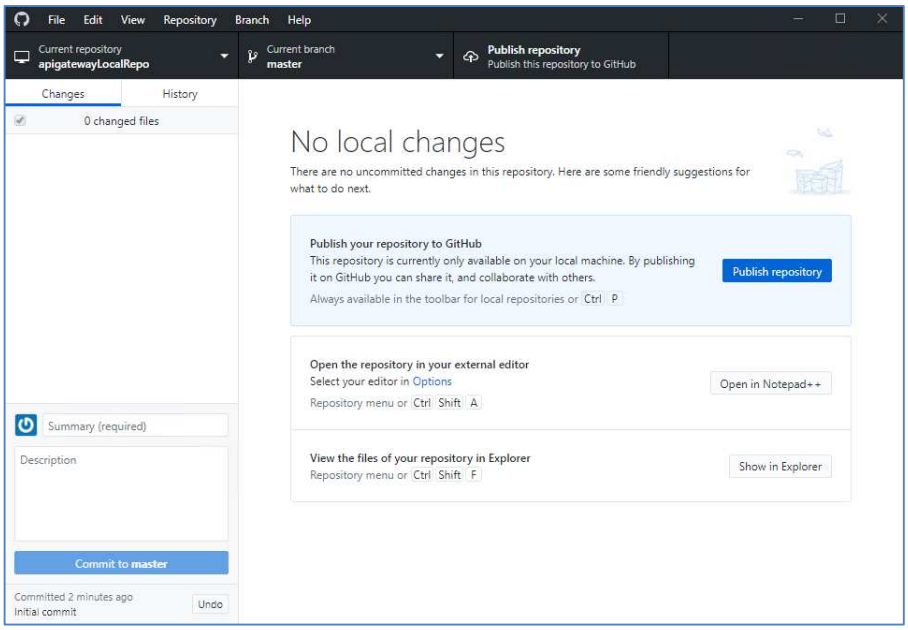
    a) **Name:** apigatewayLocalRepo

    b) **Description:** API Gateway local repository

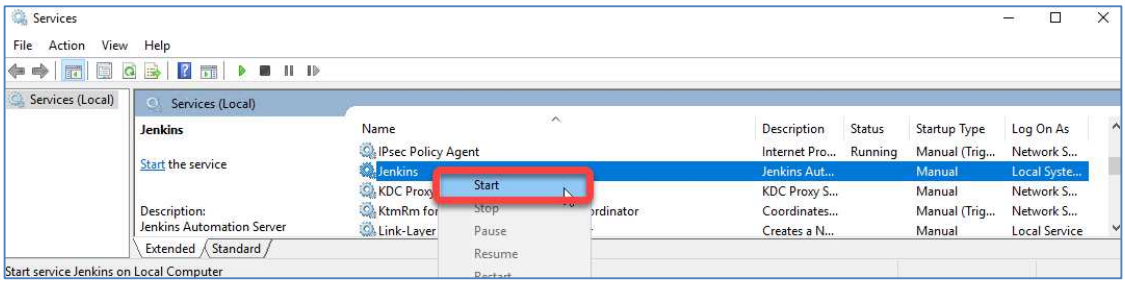
    c) **Local path:** C:\



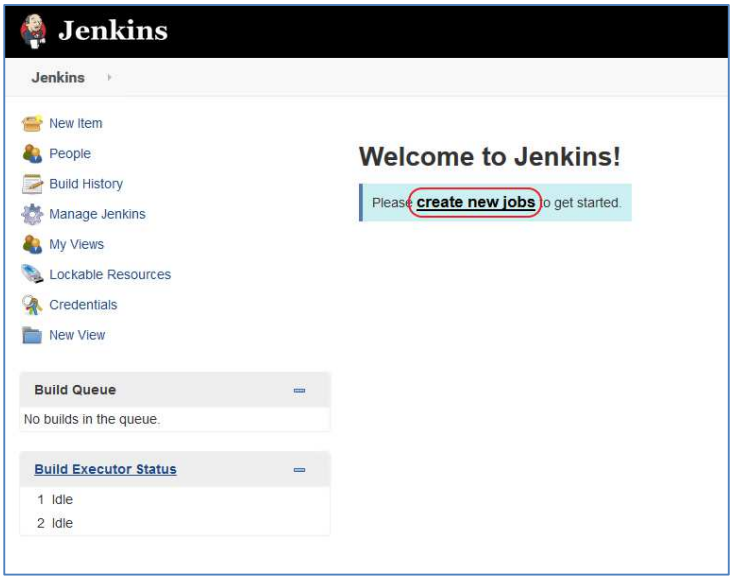

Click **Create repository**.



12. Open Windows Services panel and start **Jenkins** as a Windows service.



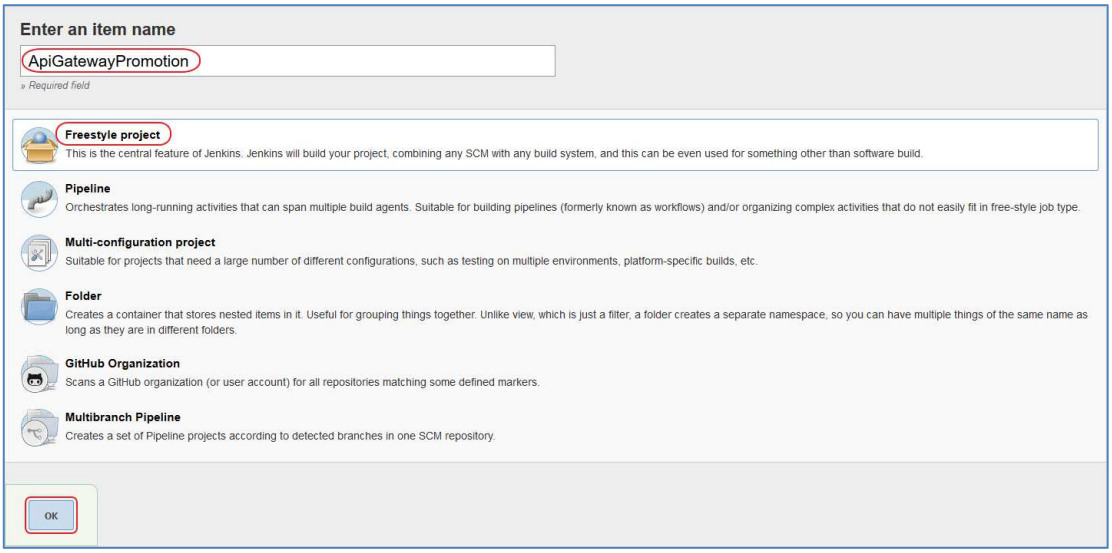

13. Login to **Jenkins** by opening the URL **http://localhost:8080** in Firefox. Login as Administrator (**Administrator/manage**).




14. Click on **create new jobs** and provide the following properties:

    a) **Enter an item name:** ApiGatewayPromotion

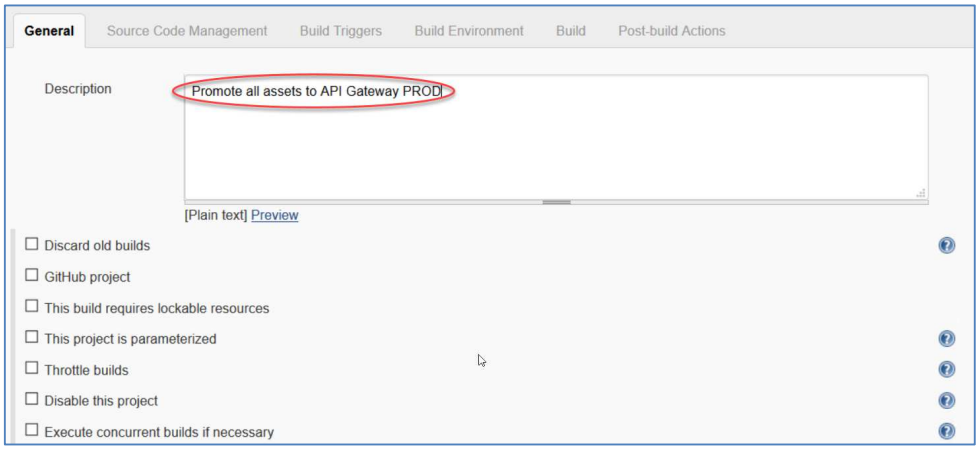
    b) Select **Freestyle project**




Click **OK**.

15. Provide the following properties:

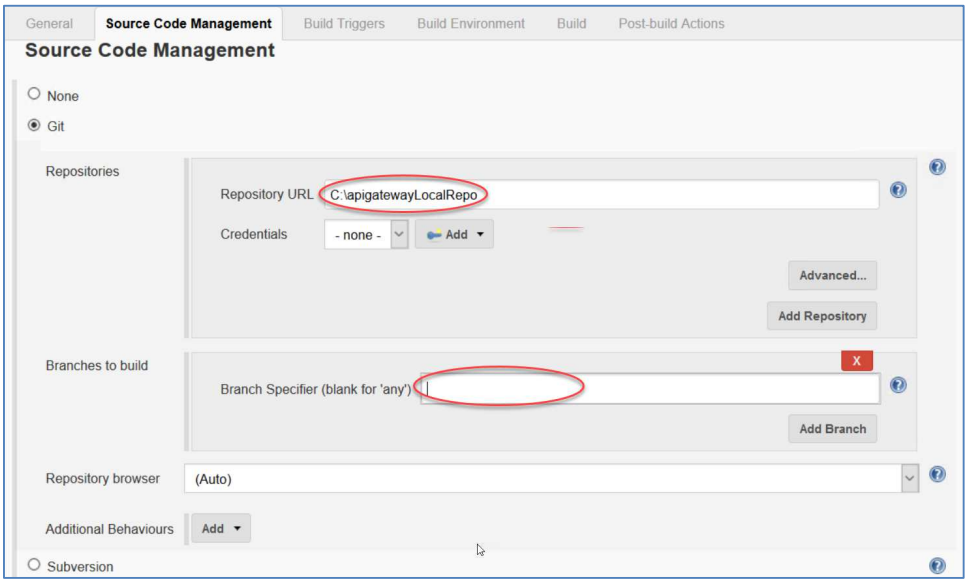
a) **Description:** Promote all assets to API Gateway PROD




b) **Source Code Management:** Git

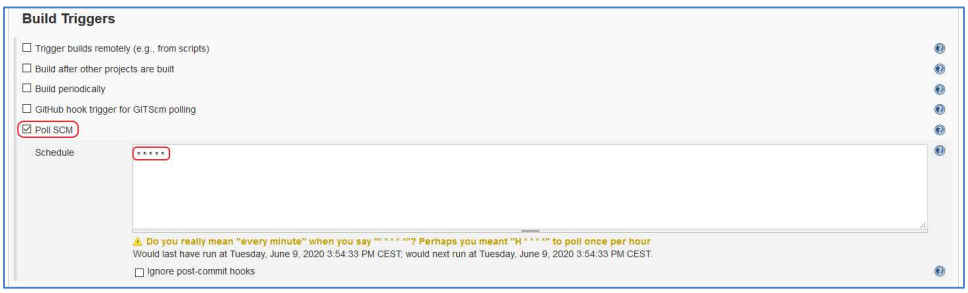
i. **Repository URL:** C:\apigatewayLocalRepo

ii. **Branch Specifier:** *< set to blank>*




c) **Build Triggers**: Poll SCM

i. **Schedule**: * * * * * (that's every minute !)



d) **Build – Add build step**: Execute Windows batch command

i. **Command (line 1):**
cd \SoftwareAG\common\AssetBuildEnvironment\bin

ii. **Command (line 2):**
build.bat -Dbuild.output.dir=C:\apigatewayRepoForDeployer\
-Dapigateway.repo.createFromLocalRepo=true
-Dapigateway.repo.localRepo.path
="C:\Program Files (x86)\Jenkins\workspace\ApiGatewayPromotion"

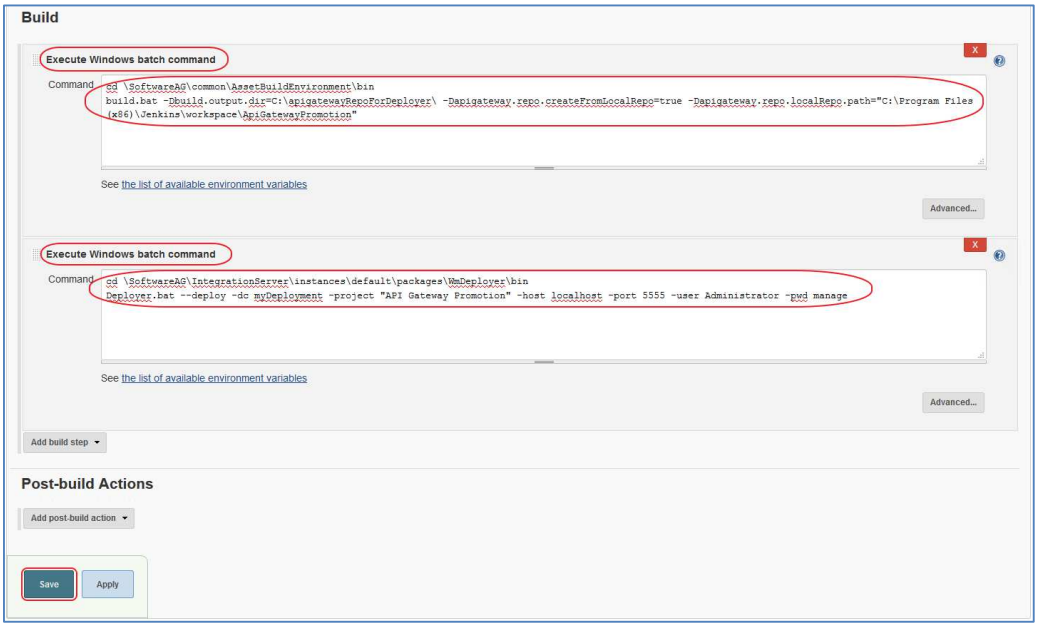
*Note*: You can copy and paste the command lines from the file **build_and_deployment.txt** in **C:\Training\E456C04-75E\Lab8**.

e) **Build – Add build step**: Execute Windows batch command

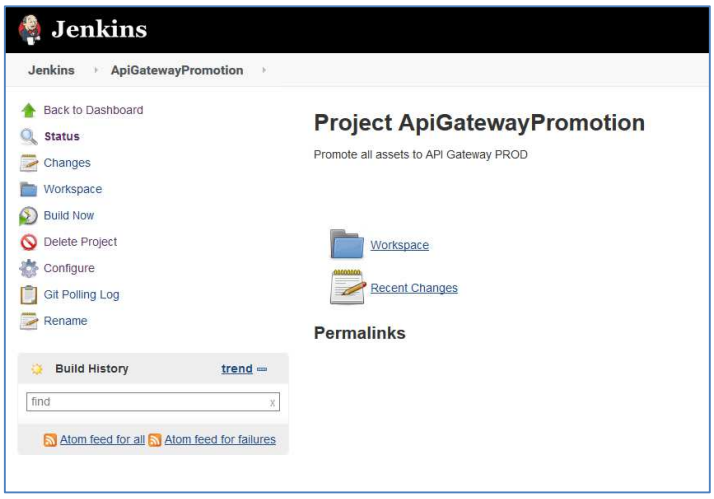
i. **Command (line 1)**:
cd \SoftwareAG\IntegrationServer\instances
\default\packages\WmDeployer\bin

ii. **Command (line 2)**:
Deployer.bat --deploy -dc myDeployment
-project "API Gateway Promotion"
-host localhost -port 5555 -user Administrator -pwd manage

*Note*: You can copy and paste the command lines from the file **build_and_deployment.txt** in **C:\Training\E456C04-75E\Lab8**.



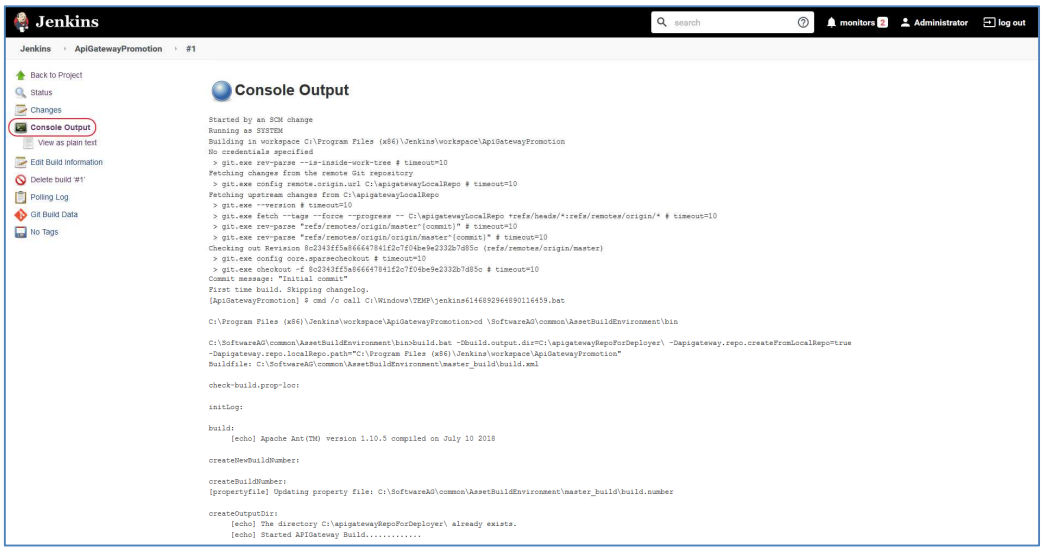
f) Click **Save**.



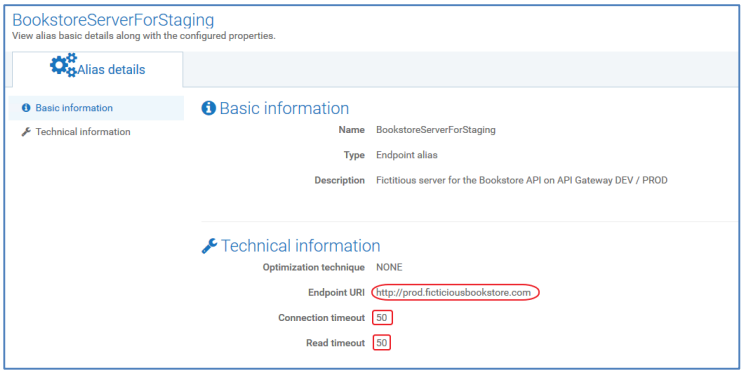

After some time (max. one minute), the job will run automatically.


And it should finish within about one minute.

g) Click on **#1** and inspect the build details including the **Console Output**.

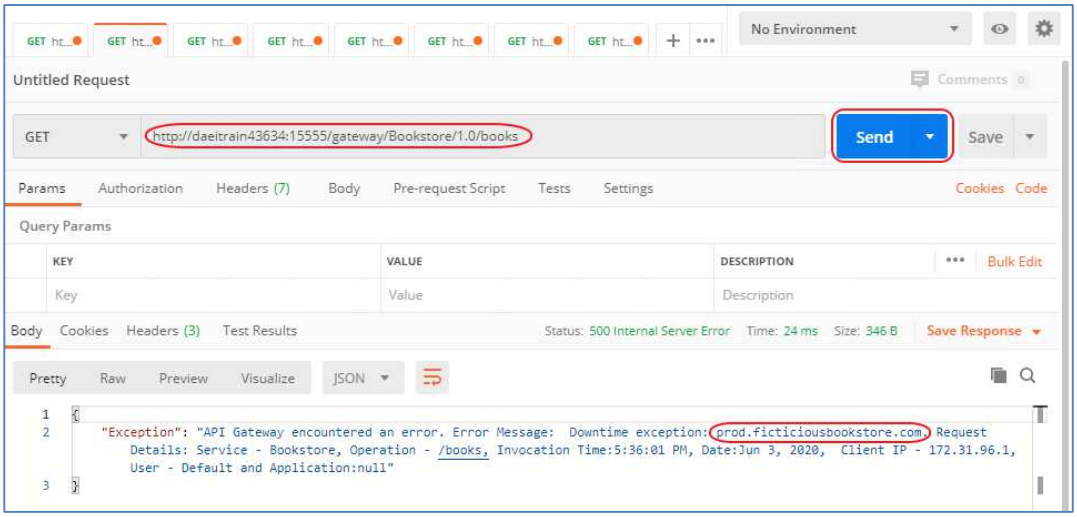



16. You might want to inspect the repository created in **C:\apigatewayRepoForDeployer**.
17. In a new private browser window, logon to **API Gateway Prod** as user **Administrator/manage**.
18. Navigate to **Administration > General > Extended settings** and check that the **pg_JWT_isHTTPS**, **pg_oauth2_isHTTPS** and **pg_OpenID_isHTTPS** settings were staged correctly.
19. Navigate to **Administration > Security > JWT/OAuth/OpenID** and check that that the **local** Authorization server alias was staged correctly including the **bookstore** OAuth2 scope.
20. Navigate to **OAuth/OpenID scopes** and check that that the **local:bookstore** scope mapping was staged correctly.
21. Navigate to **Policies > Global policies** and check that the **Throttling** global policy was staged correctly.
22. Navigate to **User management** and check that that the two user accounts were staged correctly.
23. Navigate to **APIs** and check that that the four APIs were staged correctly.
24. Navigate to **Aliases** and check that that the **BookstoreServerForStaging** alias was staged correctly. It should contain the values configured in the **webMethods Deployer** deployment map.



25. Open the **Bookstore** API and copy the gateway endpoint into the Windows clipboard.

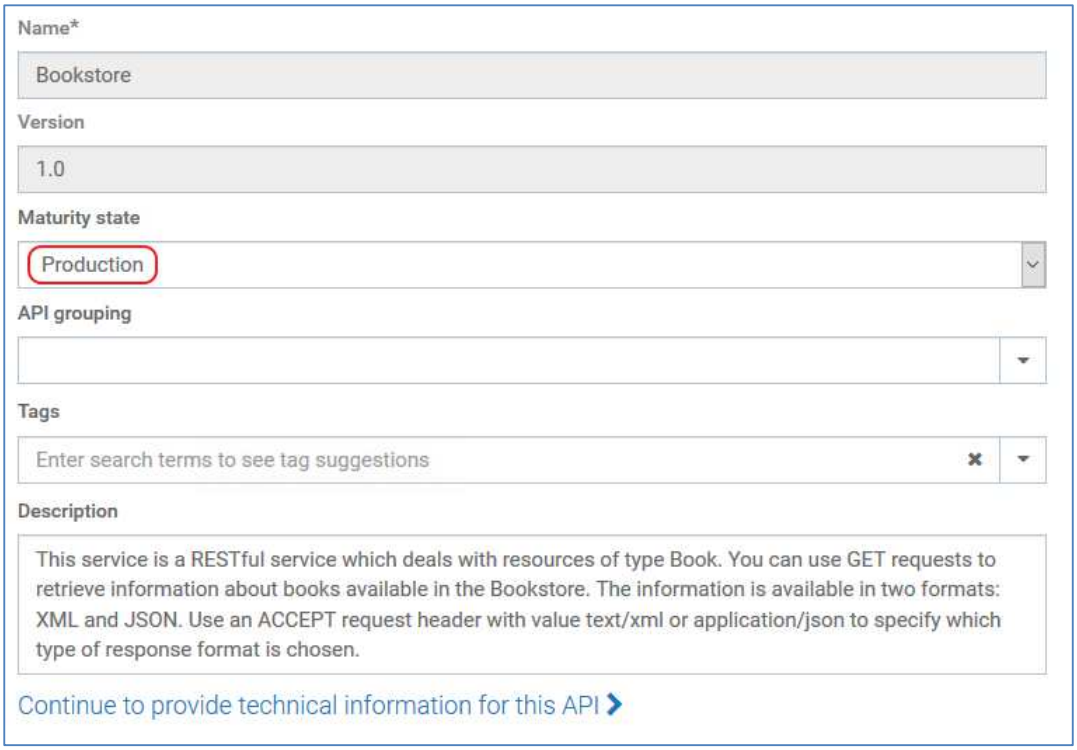
26. Open **Postman** REST client and paste the URL into the REST client and append "/**books**" to the URL; then click **Send**. This will invoke the Bookstore API on the **API Gateway Prod Stage**. As the endpoint configured in the BookstoreServerForStaging alias is invalid, the API Gateway will return an error message.




Inspect the error message and check that the **API Gateway Prod** tried to forward the request to **prod.ficticiousbookstore.com**.

27. In a new private browser window, logon to **API Gateway Dev** as user **Administrator/manage**.

28. Navigate to the **Bookstore** API, click **Edit** and change the Maturity state to **Production**.



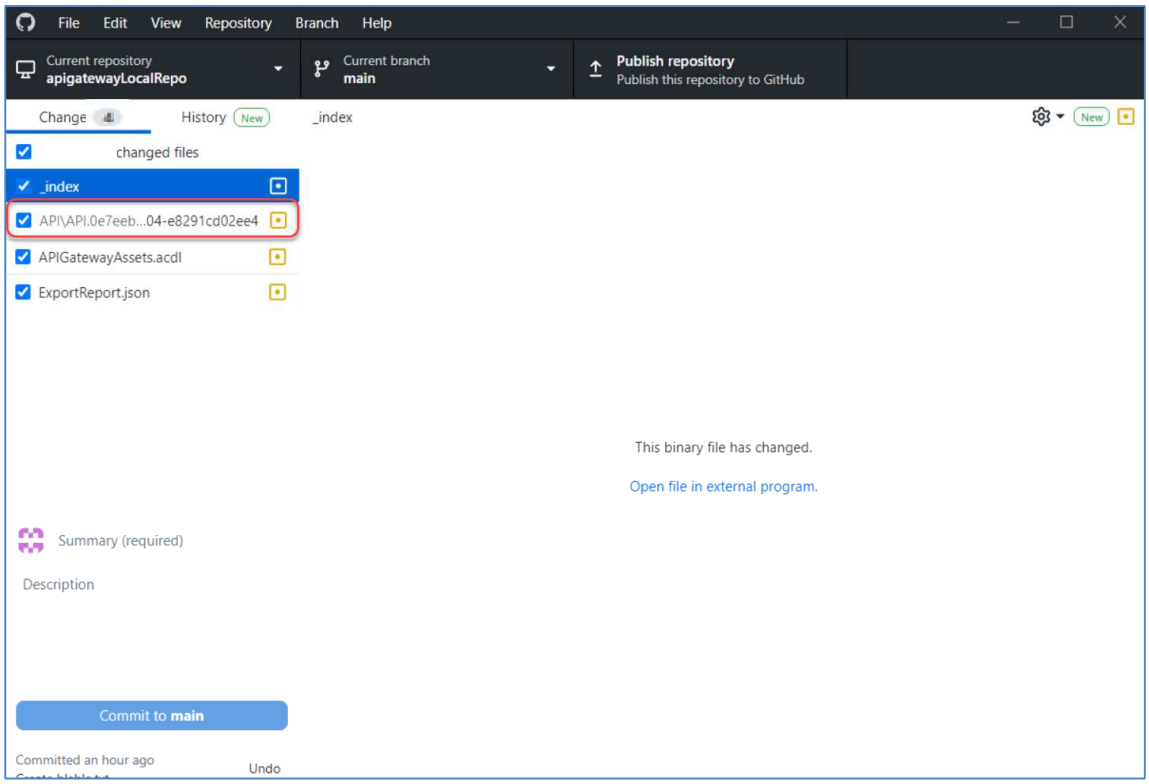

Click **Save**.

29. Back in the Windows **Command Prompt**, run the **Asset Build Environment** build script again to create a new local repository including all assets from the **API Gateway Dev** environment:

    build.bat -Dbuild.output.dir=C:\apigatewayLocalRepo\
    -Dapigateway.is.url=http://localhost:5555
    -Dapigateway.is.username=Administrator -Dapigateway.is.password=manage
    -Dapigateway.buildLocalRepoOnly=true

    *Note*: You can copy and paste the command from the file **build_and_deployment.txt** in **C:\Training\E456C04-75E\Lab8**.

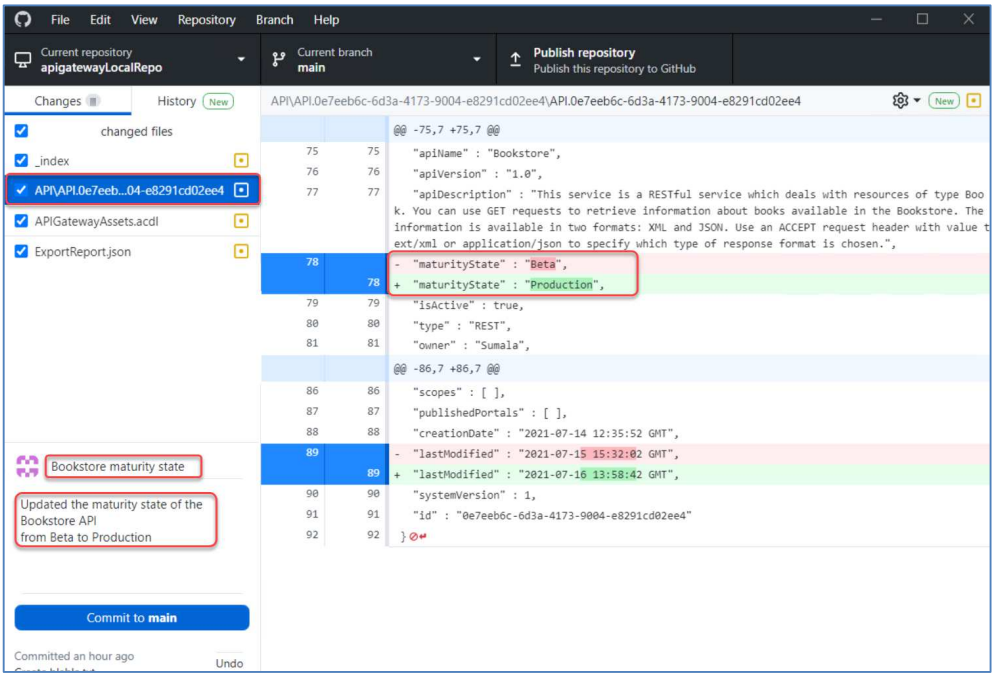
30. Switch back to the **GitHub Desktop Git** client. It has detected four file changes in the local repository.




    a) Click on the changed API definition file and inspect the change. Select all four files and provide the following properties:

        i. **Summary (required):** Bookstore maturity state

ii. **Description:** Updated the maturity state of the Bookstore API from Beta to Production




b) Click on **Commit to main**.

31. Go back to the **ApiGatewayPromotion** project in **Jenkins**. After max. one minute, it should run automatically and finish after about one minute.



32. Go back to the **API Gateway Prod** and check the **Bookstore** API. The **Maturity state** should have been updated automatically to **Production**.