

PaaS-TA App 개발

기술연구소

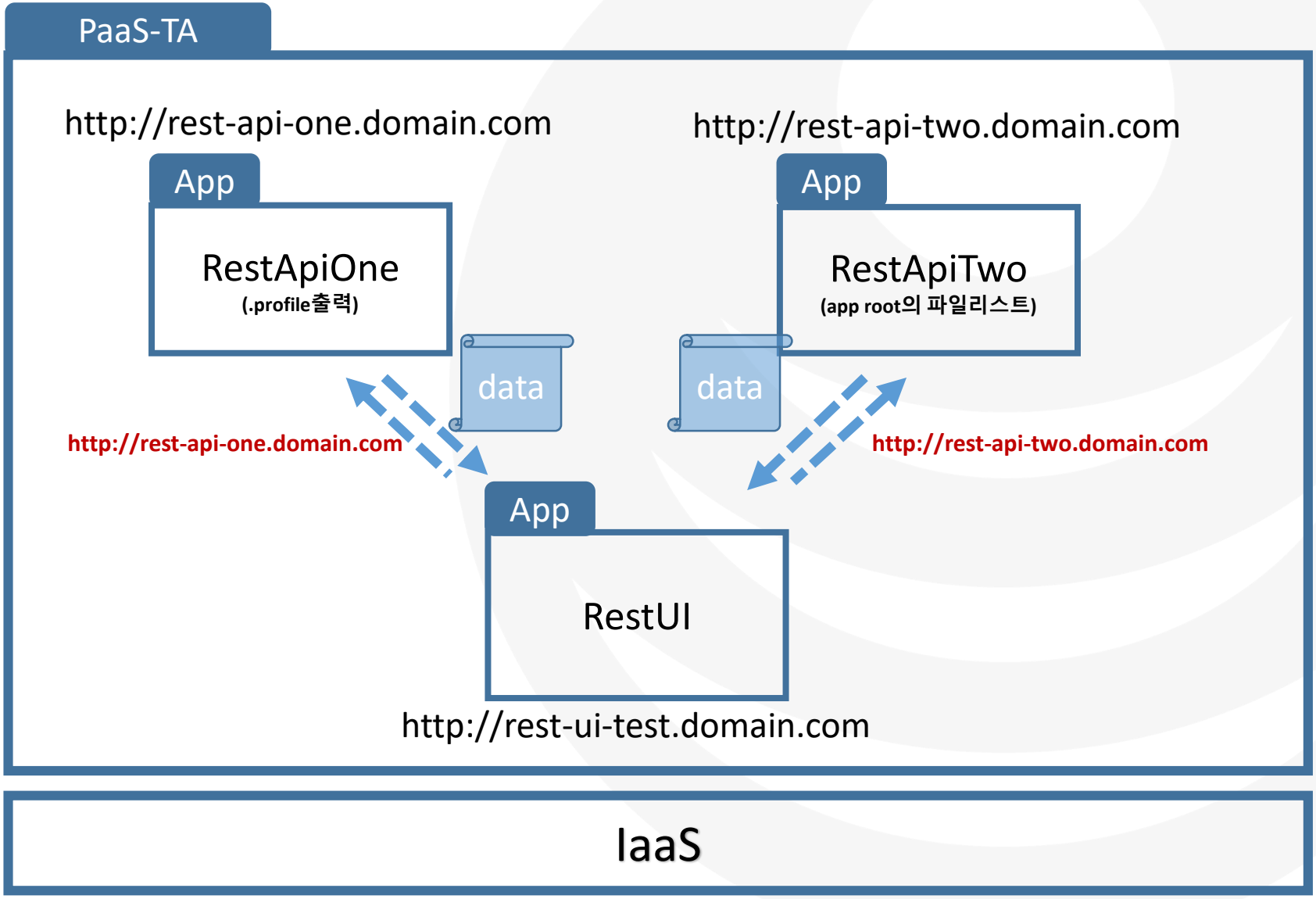
2018.11.10

1. Overview

2. IDE 선택

3. Rest App 개발

4. 배포


PaaS-TA
http://rest-api-one.domain.com
http://rest-api-two.domain.com
App
RestApiOne
(.profile출력)
App
RestApiTwo
(app root의 파일리스트)
data
data
http://rest-api-one.domain.com
http://rest-api-two.domain.com
App
RestUI
http://rest-ui-test.domain.com
IaaS

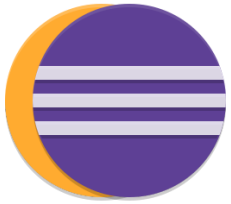
RestUI
- UI상에서 버튼을 눌러 RestApiOne, RestApiTwo를 호출합니다.
- API로 부터 얻은 정보를 화면에 출력합니다.

RestApiOne
- http 호출에 의해 요청이 오면 특정 디렉토리에 있는 파일을 열어 응답합니다.

RestApiTwo
- http 호출에 의해 요청이 오면 App root경로의 파일 리스트를 보여줍니다.

2. IDE 선택

- 어떤 IDE를 사용해도 무방하나 최신버전으로 사용하는것을 권고합니다.
- 개발 편의성을 고려하여 REST프로젝트에 맞는 템플릿이 제공되는 IDE를 추천합니다.
- 본 개발 설명에서 주로 사용할 기술인 SpringBoot 디렉토리 구조를 쉽게 만들어주는 템플릿기능을 제공하는 STS 혹은 STS plugin을 활용하도록 하겠습니다.
- STS(Spring Tool Suits)는 sprin개발에 특화된 IDE로 많은 기본 templat을 제공합니다.
- Eclipse 또한 어느버전이든 상관없지만, STS plugin을 사용할 수 있는 버전을 권장합니다. (이전버전이라고 하더라도 디렉토리 구조를 수동으로 생성해도 됩니다.)

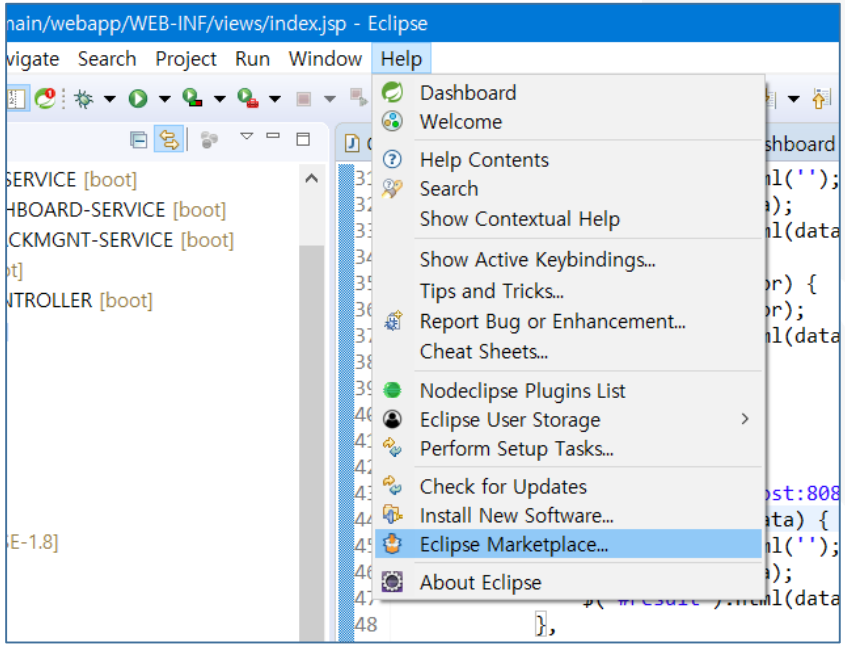




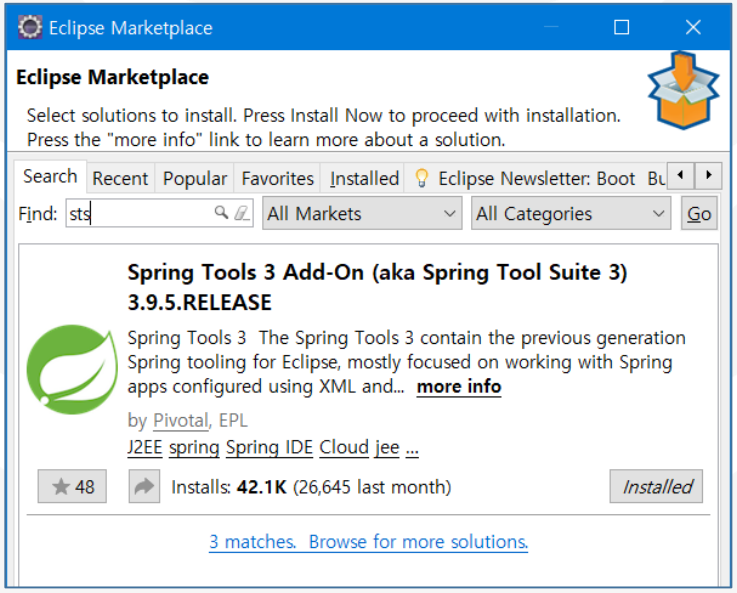
eclipse

아래 링크를 통해 최신버전을 다운로드 할 수 있습니다.
https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2018-09/Ra/eclipse-inst-win64.exe

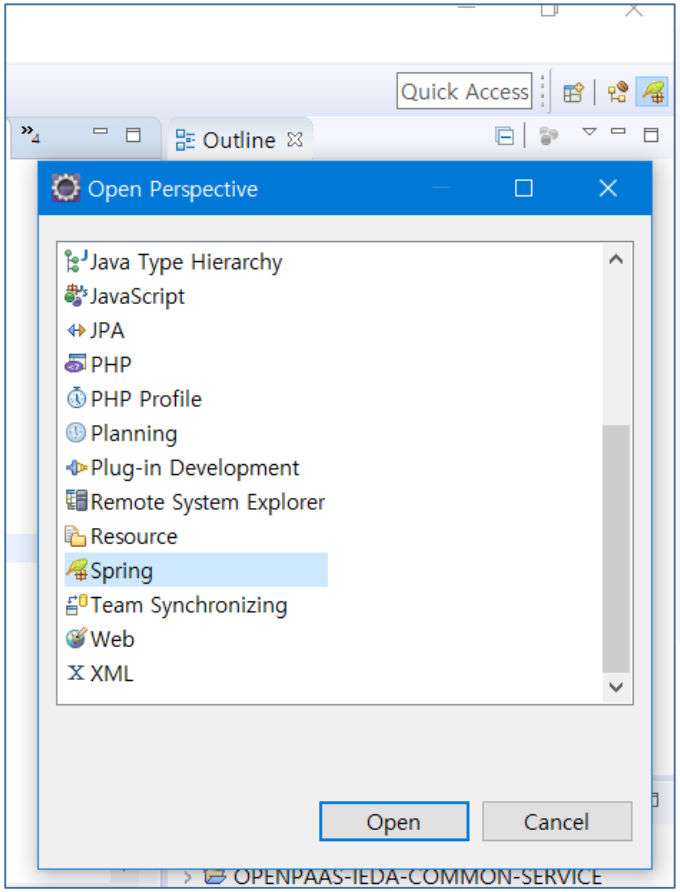
1. Help > Eclipse Marketplace… 클릭



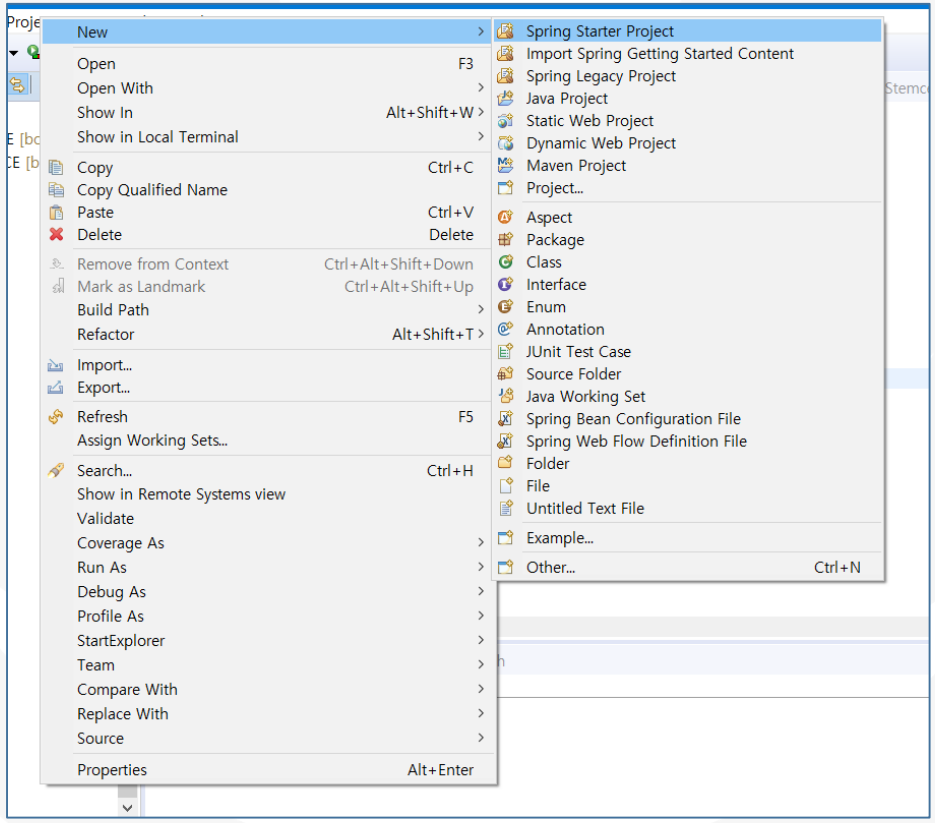
2. sts 검색 후 install 클릭



3. 우측상단의 Open Perspective 클릭



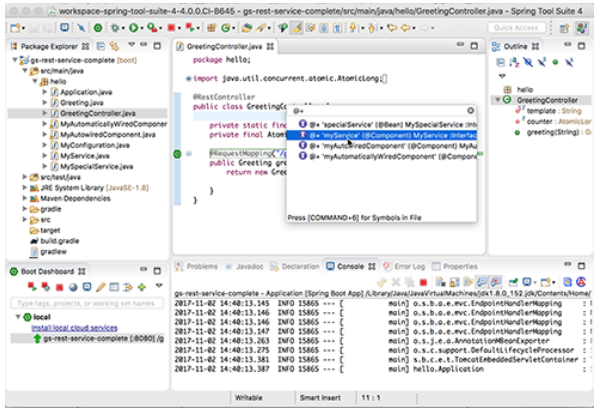
4. Spring Starter Project 사용 가능

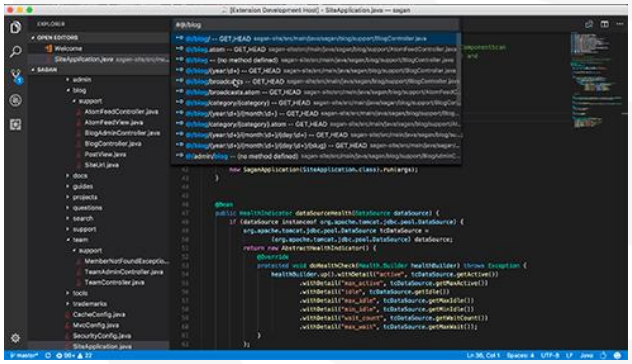


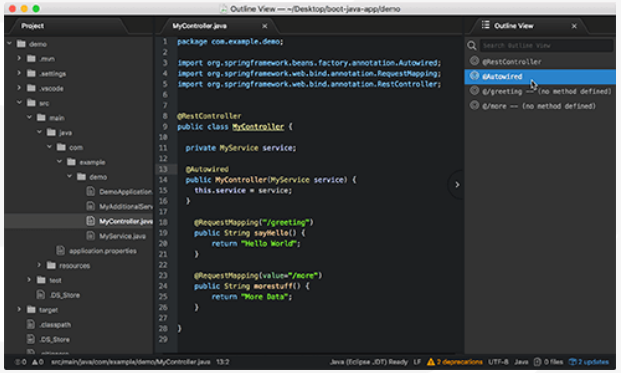
2. IDE 선택 - STS

STS4

아래 링크를 통해 최신버전을 다운로드 할 수 있습니다.
https://spring.io/tools
(Eclipse, VSCode, atom 버전으로 제공)



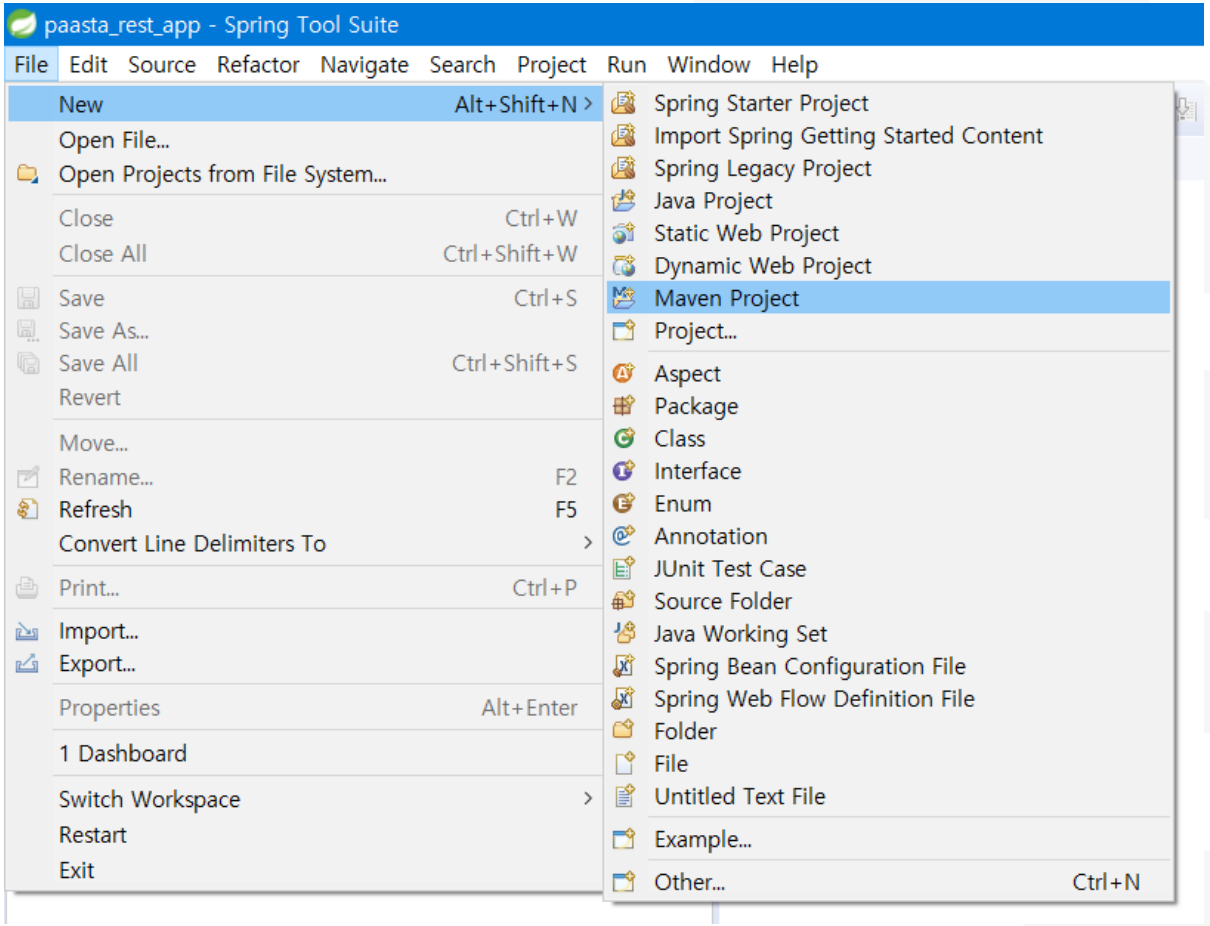




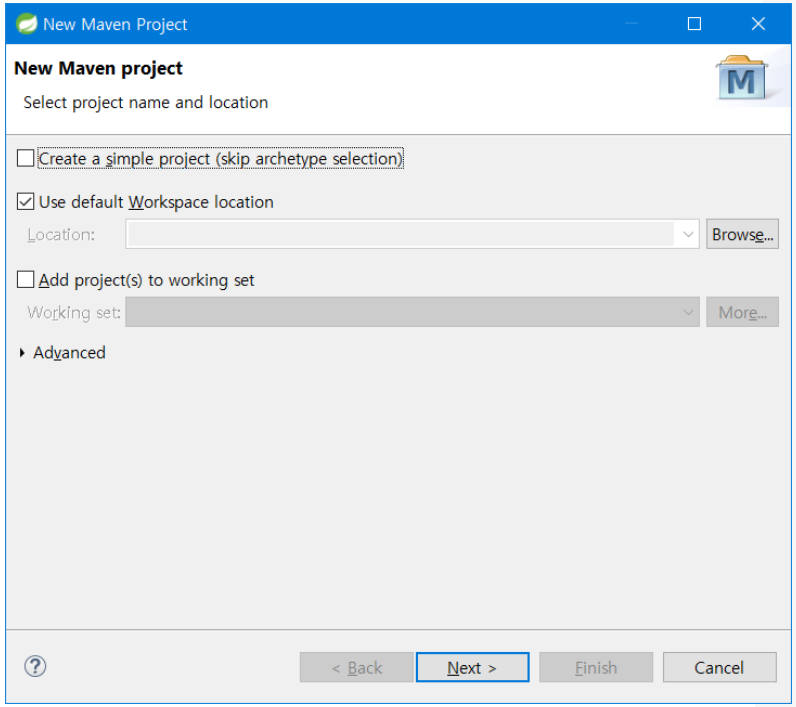
RestUI

1. 프로젝트 생성

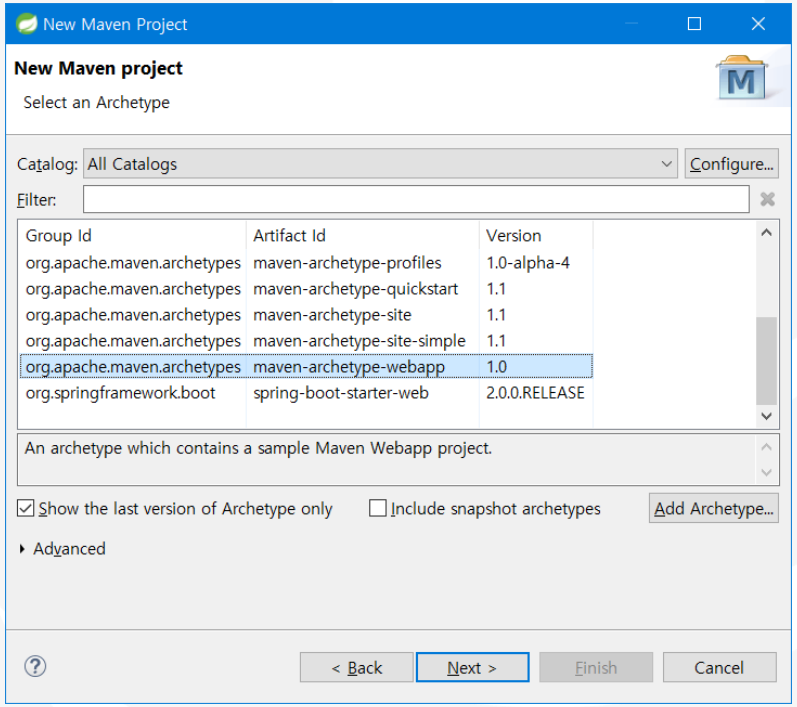
File > New > Maven Project



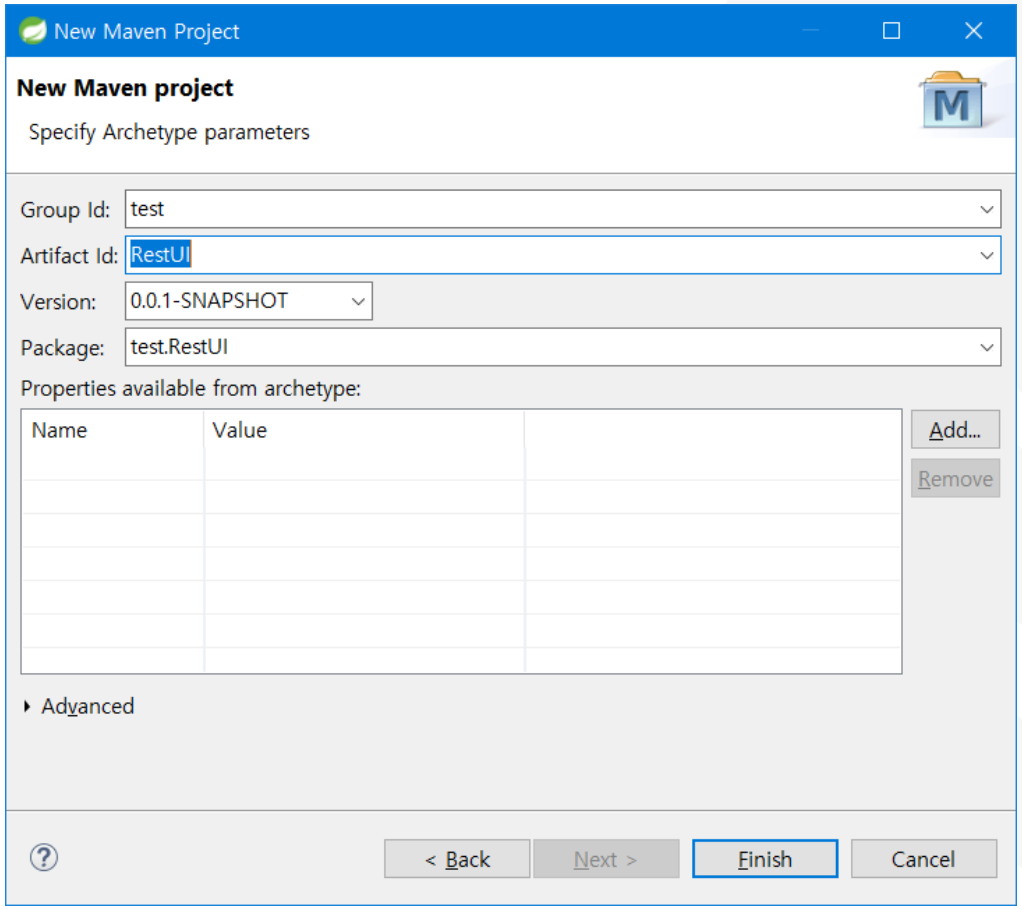
Next > 클릭



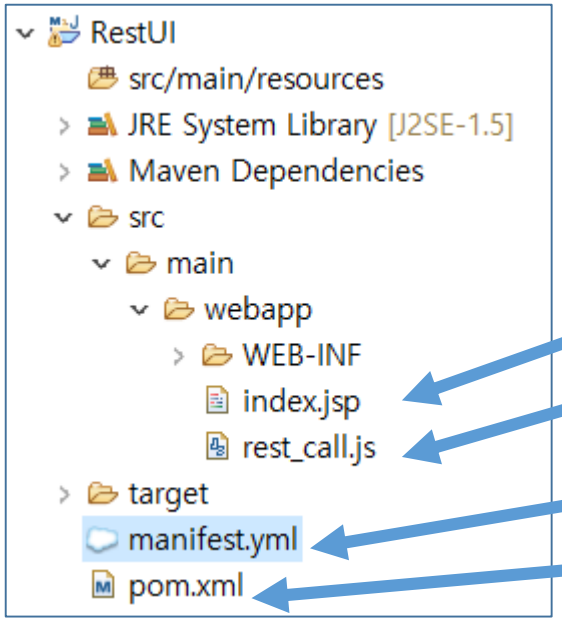
Maven-archetype-webapp 선택



Group Id : test
Artifact Id : RestUI(프로젝트 이름)
Finish 클릭



디렉토리 구조



Rest 호출 페이지

Ajax로 Rest호출하기 위한 javascript

배포를 위한 배포 명세서 manifest.yml

Dependency정의 명세서 pom.xml

pom.xml에 dependency 추가

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>test</groupId>
  <artifactId>RestUI</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>RestUI Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
    </dependency>

  </dependencies>
  <build>
    <finalName>RestUI</finalName>
  </build>
</project>
```

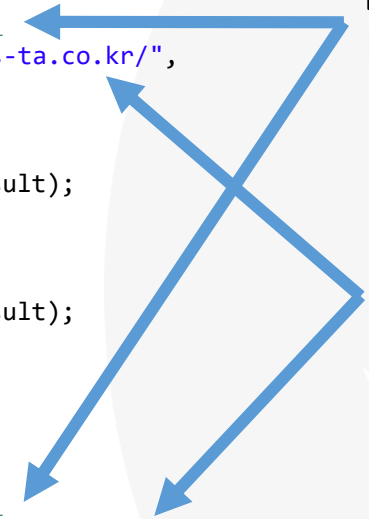
jsp 파일 구동을 위한 dependency

index.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <title>PaasTa</title>
    <script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
    <script type="text/javascript" src="rest_call.js"></script>
</head>
<body>
    <div>
        <button onclick="javascript:go1();">API_ONE</button>
        <button onclick="javascript:go2();">API_TWO</button>
    </div>
    <div id="result">
    </div>
</body>
</html>
```

rest_call.js

```
function go1() {
    $.ajax({
//      url: "http://localhost:8081/",
        url: "http://rest-api-one.paas-ta.co.kr/",
        success: function(data) {
        $("#result").html('');
            console.log(data);
            $("#result").html(data.result);
        },
        error: function(error) {
            console.log(error);
            $("#result").html(data.result);
        }
    });
}
function go2() {
    $.ajax({
//      url: "http://localhost:8082/",
        url: "http://rest-api-two.paas-ta.co.kr/",
        success: function(data) {
        $("#result").html('');
            console.log(data);
            $("#result").html(data.result);
        },
        error: function(error) {
            console.log(error);
            $("#result").html(data.result);
        }
    });
}
```

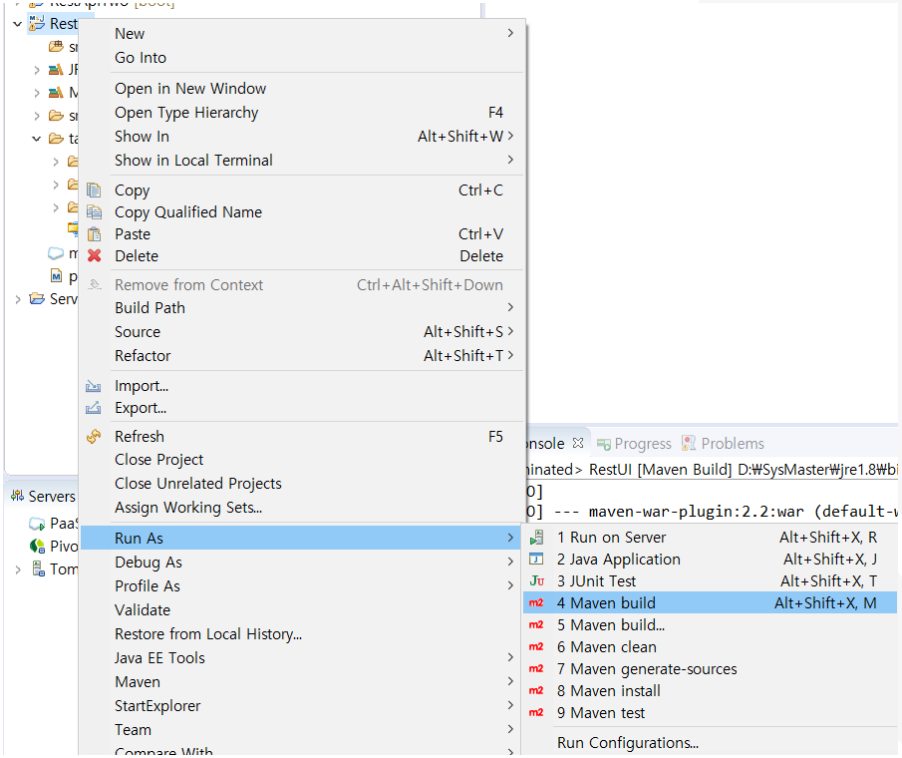


Localtest용 URL SpringBoot로 Port를 다르게 테스트 할 수 있습니다.

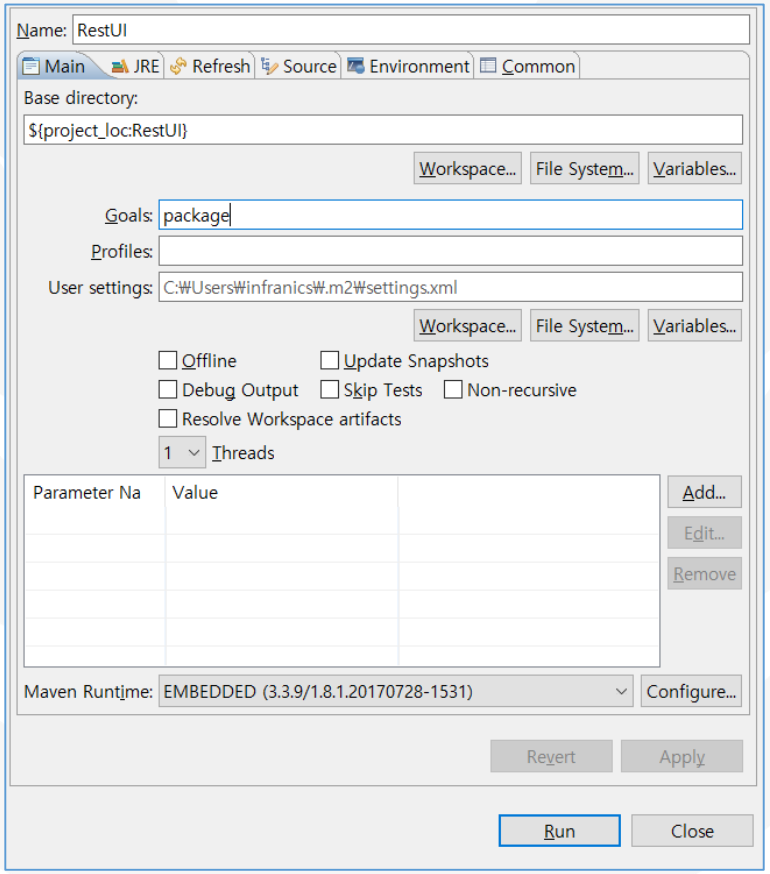
실제 RestApiOne, RestApiTwo는 각각 배포 되므로 별도의 URL을 갖습니다.

build

프로젝트 선택 > Run As > Maven build



Goals 에 package입력 > Run 클릭

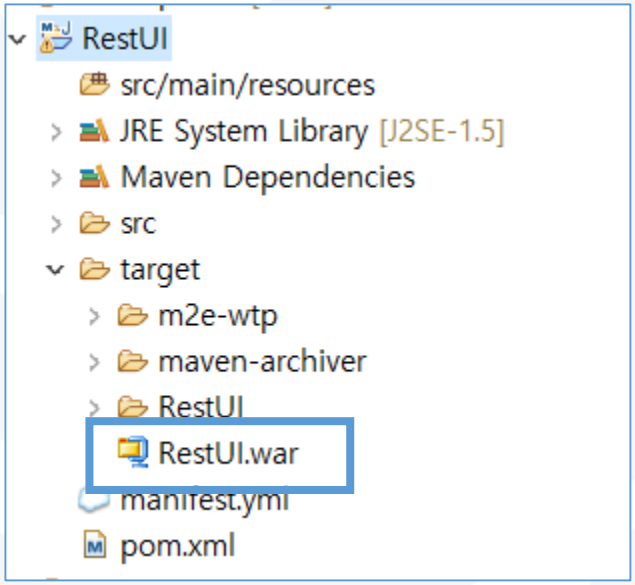


build

Maven Building

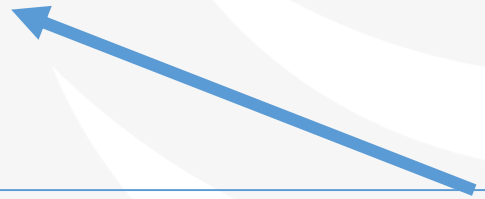
```
Console  Progress  Problems
<terminated> RestUI [Maven Build] D:₩SysMaster₩jre1.8₩bin₩javaw.exe (2018. 11. 12. 오후 7:54:16)
[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ RestUI ---
[INFO] Packaging webapp
[INFO] Assembling webapp [RestUI] in [D:\test\paasta_rest_app\RestUI\target\RestUI]
[INFO] Processing war project
[INFO] Copying webapp resources [D:\test\paasta_rest_app\RestUI\src\main\webapp]
[INFO] Webapp assembled in [52 msecs]
[INFO] Building war: D:\test\paasta_rest_app\RestUI\target\RestUI.war
[INFO] WEB-INF\web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1.443 s
[INFO] Finished at: 2018-11-12T19:54:18+09:00
[INFO] Final Memory: 8M/20M
[INFO] ------------------------------------------------------------------------
```

Target > RestUI.war생성 완료



manifest.yml

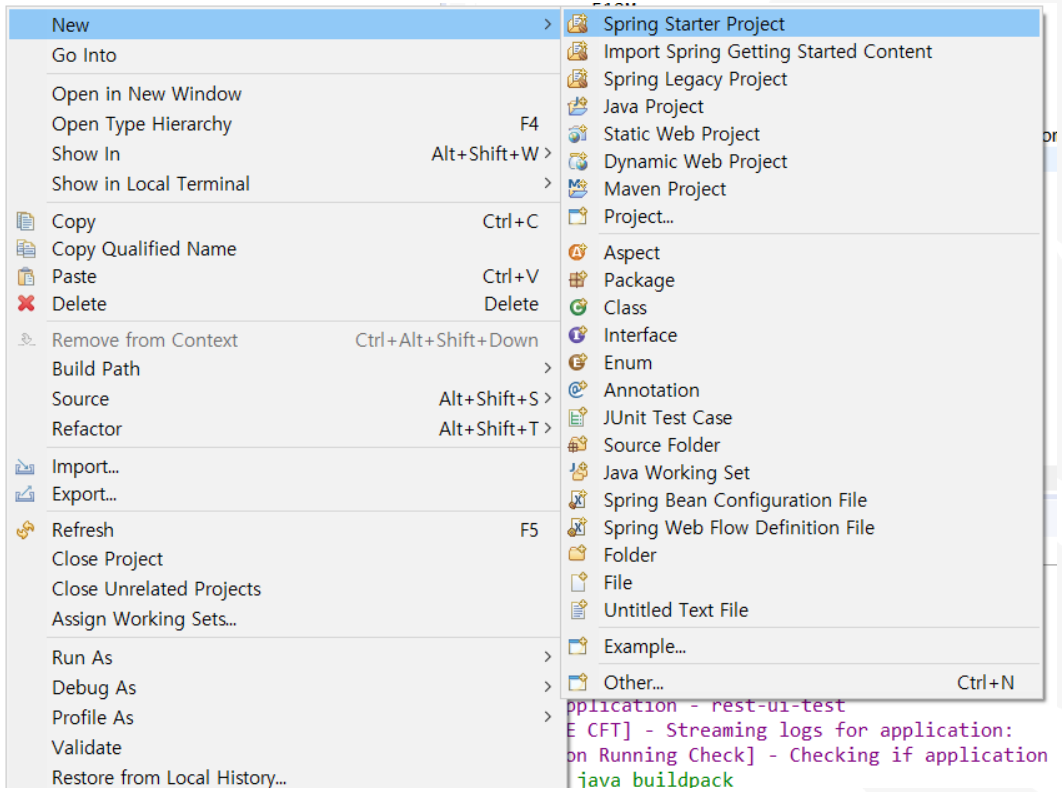
```
---
applications:
- name: rest-ui-test
  memory: 512M
  host: rest-ui-test
  path: ./target/RestUI.war
  buildpack: java_buildpack
env:
  JAVA_OPTS: -Djava.security.egd=file:///dev/urandom
```



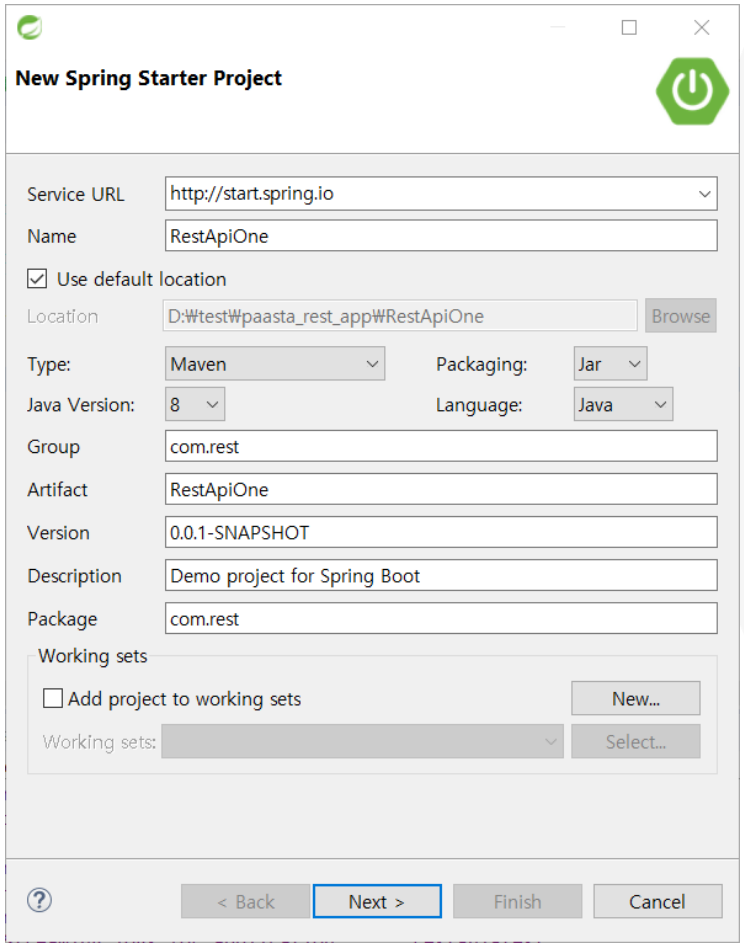
JVM에서 session을 생성할때 machine의 난수생성기의 waiting을 줄이기 위한 방법

3. Rest App 개발 – RestApiOne

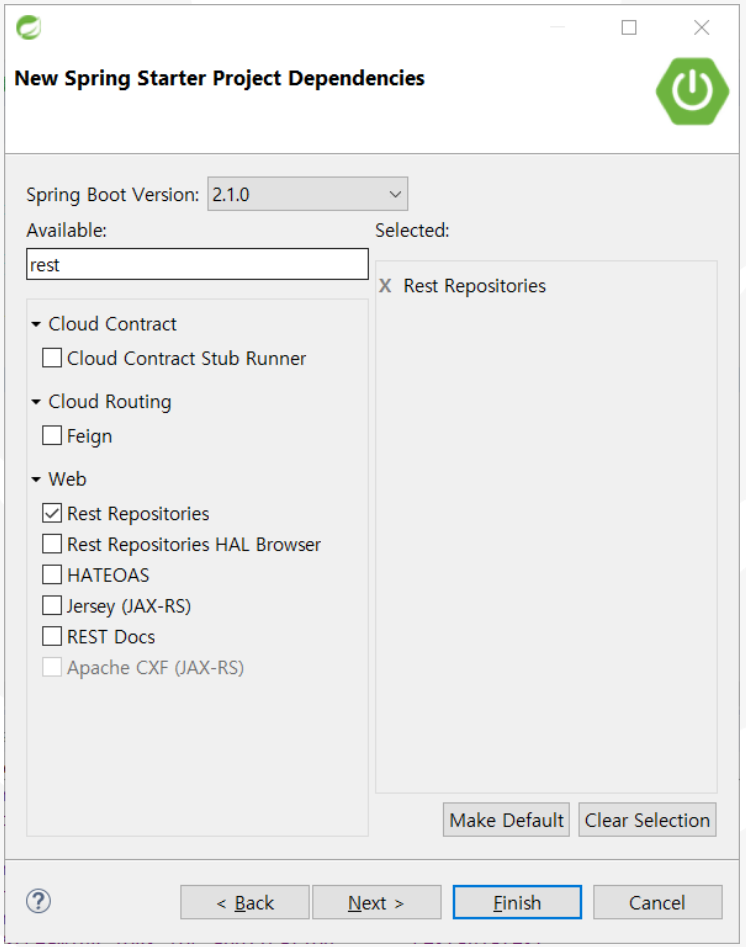
File > New > Spring Starter Project



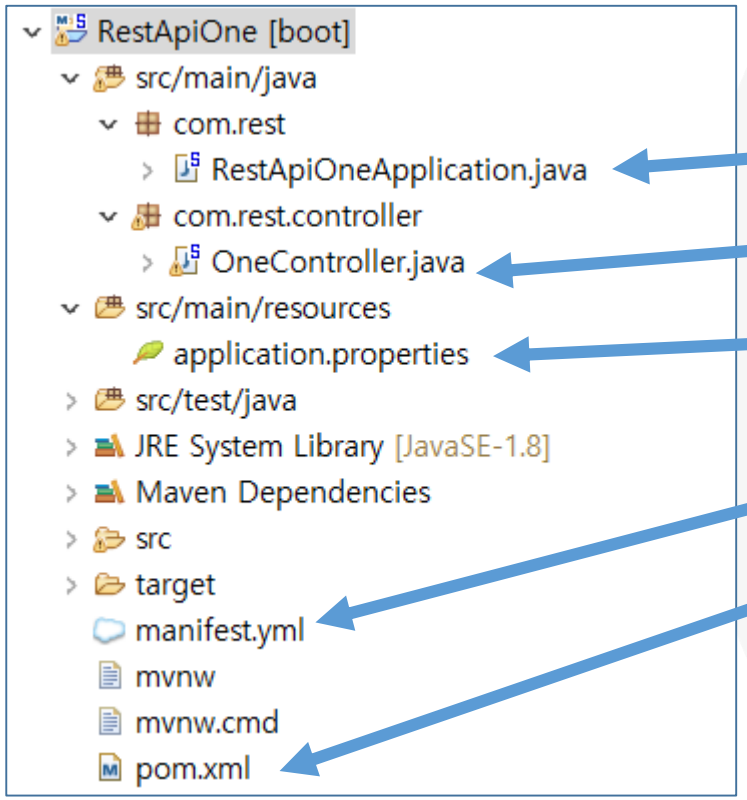
프로젝트 이름 및 패키지 등 설정



사용할 dependency 설정 : Rest Repositories



디렉토리 구조



Spring Boot 구동 (자동생성)

Rest 응답 controller

localtest시 설정값

배포를 위한 배포 명세서 manifest.yml

Dependency정의 명세서 pom.xml

pom.xml (1/2) – 자동으로 생성

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<groupId>com.rest</groupId>
<artifactId>RestApiOne</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>RestApiOne</name>
<description>Demo project for Spring Boot</description>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.1.0.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-rest</artifactId>
</dependency>
```

pom.xml (2/2) – 자동으로 생성

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>


</project>
```

application.properties

```
server.port=8081
```

Localtest를 위해 port를 지정해준다.

OneController.java

```
package com.rest.controller;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.Map;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class OneController {
	@CrossOrigin
	@RequestMapping("/")
	public Map home() throws IOException, InterruptedException {
		Runtime runtime = Runtime.getRuntime();
		Process process = runtime.exec("/bin/cat /home/vcap/.profile");
		Map rs = new HashMap();
		rs.put("result", printStream(process));
		return rs;
	}

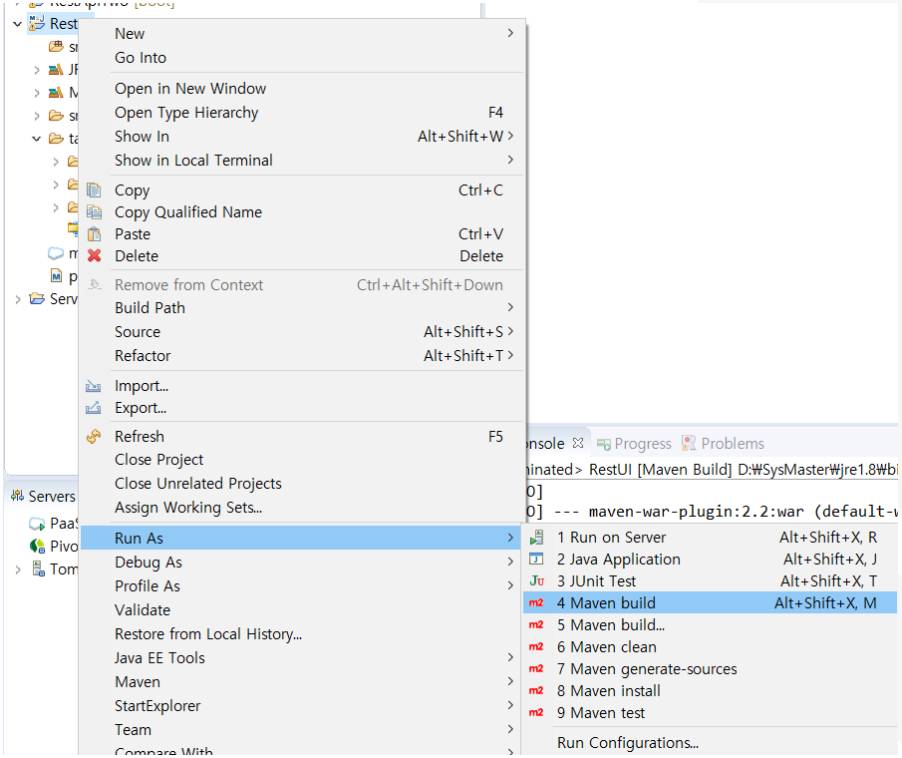
	private String printStream(Process process)  throws IOException, InterruptedException {
		process.waitFor();
		String str = "";
		BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
		String s;
		while( (s = reader.readLine()) != null ) {
			str += s+"</br>";
		}
		return str;
	}
}
```

URL이 호출되면 .profile을 읽어 result라는 이름으로 response 합니다.

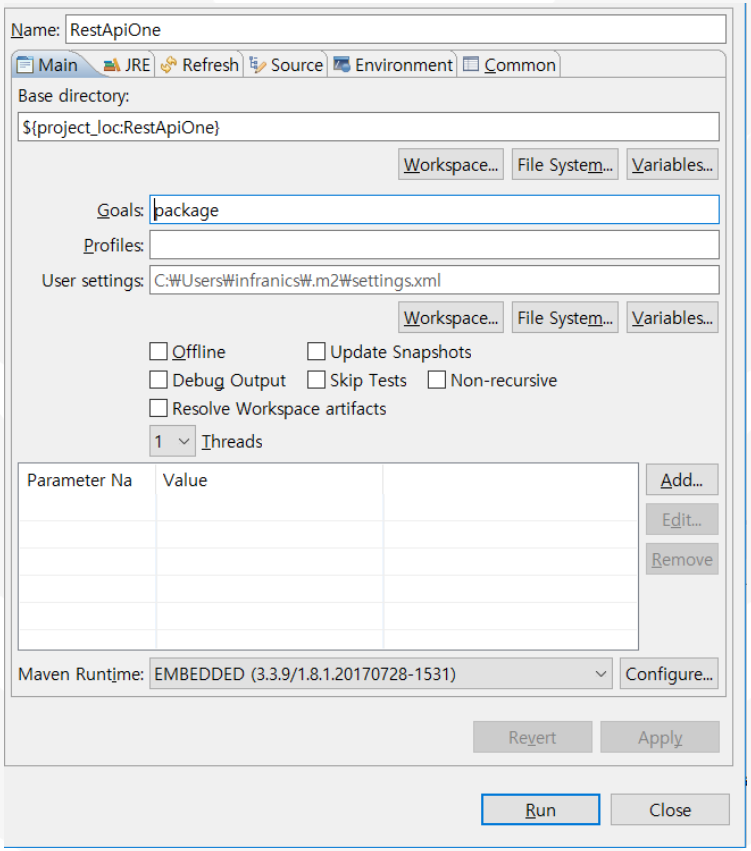
Local에서 테스트할 경우 OS에 맞는 호출로 변경하여 테스트합니다.

build

프로젝트 선택 > Run As > Maven build



Goals 에 package입력 > Run 클릭



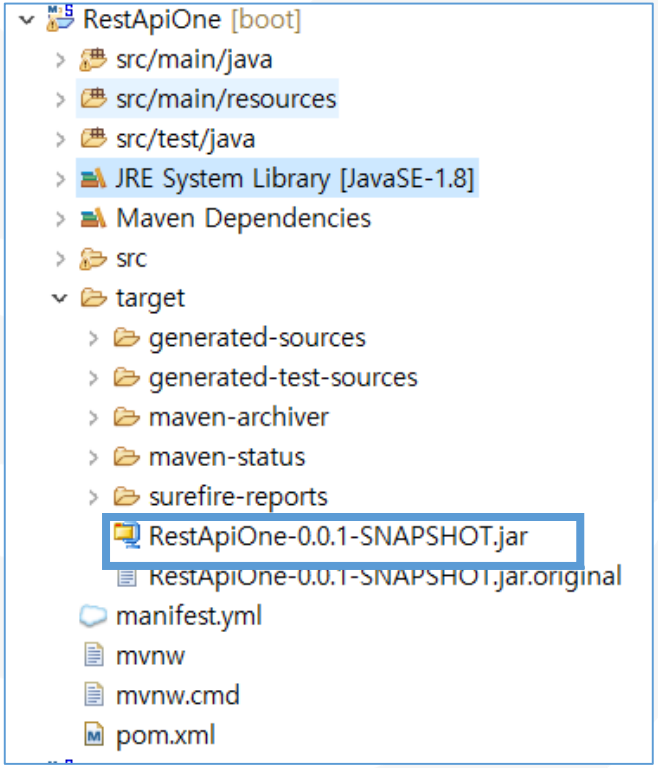
3. Rest App 개발 – RestApiOne

build

Maven Building

```
Console  Progress  Problems
<terminated> RestApiOne [Maven Build] D:₩SysMaster₩jre1.8₩bin₩javaw.exe (2018. 11. 12. 오후 7:56:18)
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.0:jar (default-jar) @ RestApiOne ---
[INFO] Building jar: D:\test\paasta_rest_app\RestApiOne\target\RestApiOne-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.1.0.RELEASE:repackage (repackage) @ RestApiOne ---
[INFO] Replacing main artifact D:\test\paasta_rest_app\RestApiOne\target\RestApiOne-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 9.382 s
[INFO] Finished at: 2018-11-12T19:56:29+09:00
[INFO] Final Memory: 19M/66M
[INFO] ------------------------------------------------------------------------
```

Target > RestApiOne-0.0.1-SNAPSHOT.jar 생성

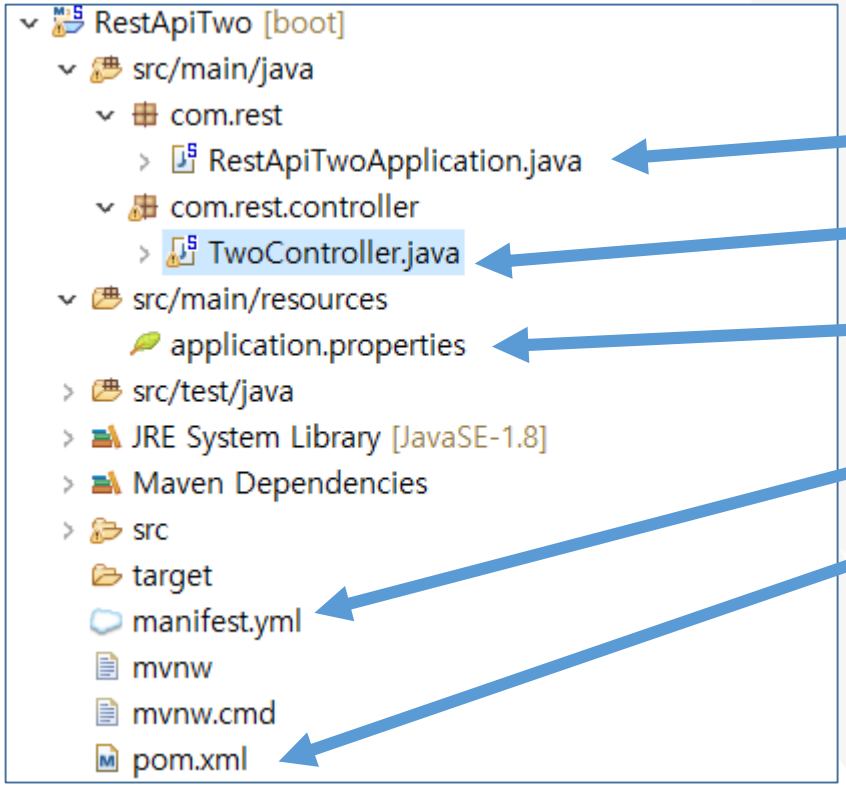


manifest.yml

```
---
applications:
- name: rest-api-one
  memory: 1G
  host: rest-api-one
  path: ./target/RestApiOne-0.0.1-SNAPSHOT.jar
  buildpack: java_buildpack
env:
  JAVA_OPTS: -Djava.security.egd=file:///dev/urandom
```

3. Rest App 개발 – RestApiTwo

RestApiOne과 동일하게 프로젝트를 생성하고 OenController.java만 다릅니다.
디렉토리 구조



Spring Boot 구동 (자동생성)

Rest 응답 controller

localtest시 설정값 : localtest용 port 8082

배포를 위한 배포 명세서 manifest.yml

Dependency정의 명세서 pom.xml

OneController.java

```
package com.rest.controller;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.Map;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class TwoController {
	@CrossOrigin
	@RequestMapping("/")
	public Map home() throws IOException, InterruptedException {
		Runtime runtime = Runtime.getRuntime();
		Process process = runtime.exec("/bin/ls -al");
		Map rs = new HashMap();
		rs.put("result", printStream(process));
		return rs;
	}

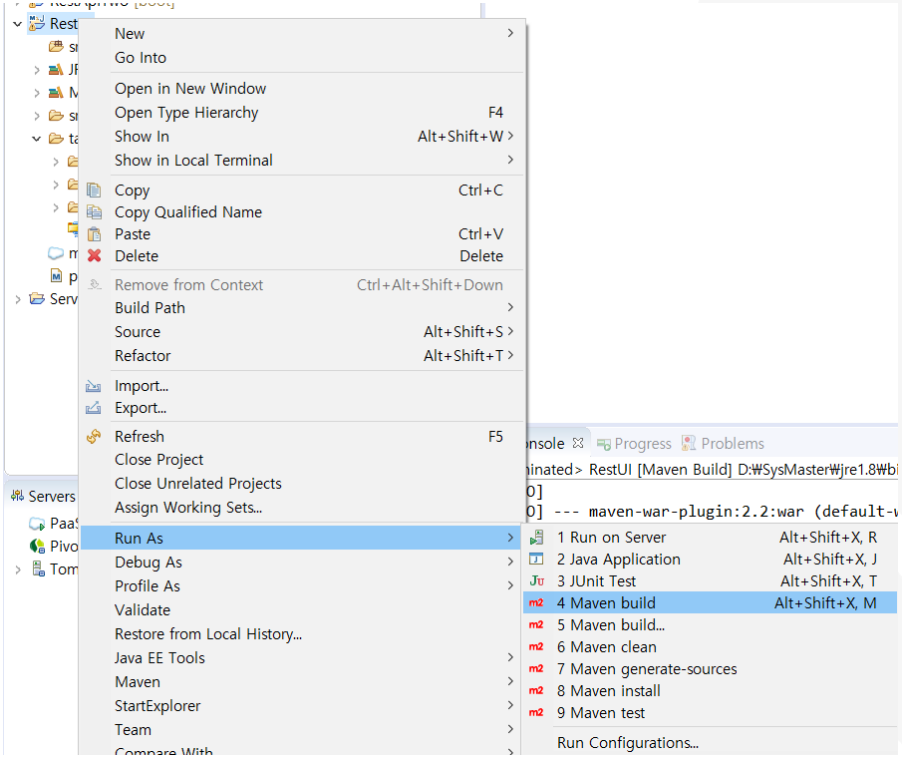
	private String printStream(Process process)  throws IOException, InterruptedException {
		process.waitFor();
		String str = "";
		BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
		String s;
		while( (s = reader.readLine()) != null ) {
			str += s+"</br>";
		}
		return str;
	}
}
```

URL이 호출되면 app의 Root의 디렉토리 파일 리스트를 읽어 result라는 이름으로 response 합니다.

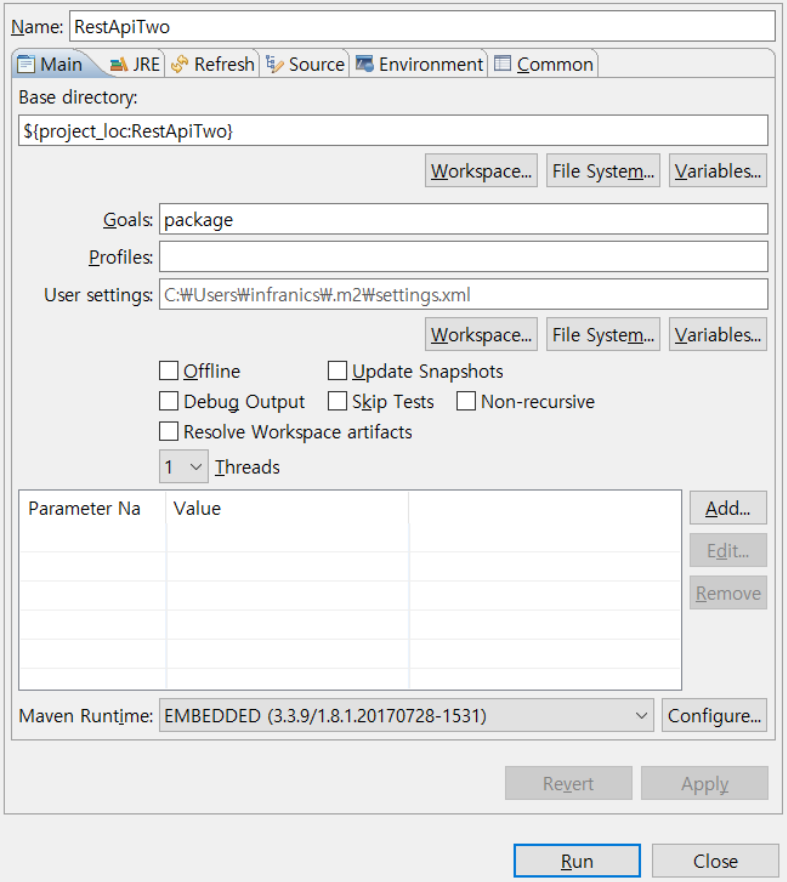
Local에서 테스트할 경우 OS에 맞는 호출로 변경하여 테스트합니다.

build

프로젝트 선택 > Run As > Maven build



Goals 에 package입력 > Run 클릭

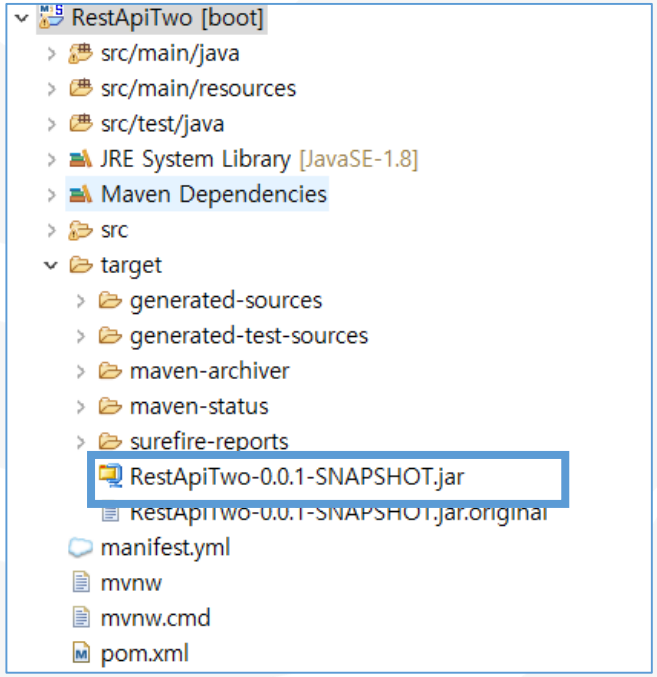


build

Maven Building

```
Console  Progress  Problems
<terminated> RestApiTwo [Maven Build] D:₩SysMaster₩jre1.8₩bin₩javaw.exe (2018. 11. 12. 오후 7:59:26)
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.0:jar (default-jar) @ RestApiTwo ---
[INFO] Building jar: D:\test\paasta_rest_app\RestApiTwo\target\RestApiTwo-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.1.0.RELEASE:repackage (repackage) @ RestApiTwo ---
[INFO] Replacing main artifact D:\test\paasta_rest_app\RestApiTwo\target\RestApiTwo-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 8.462 s
[INFO] Finished at: 2018-11-12T19:59:36+09:00
[INFO] Final Memory: 19M/49M
[INFO] ------------------------------------------------------------------------
```

Target > RestApiTwo-0.0.1-SNAPSHOT.jar 생성



manifest.yml

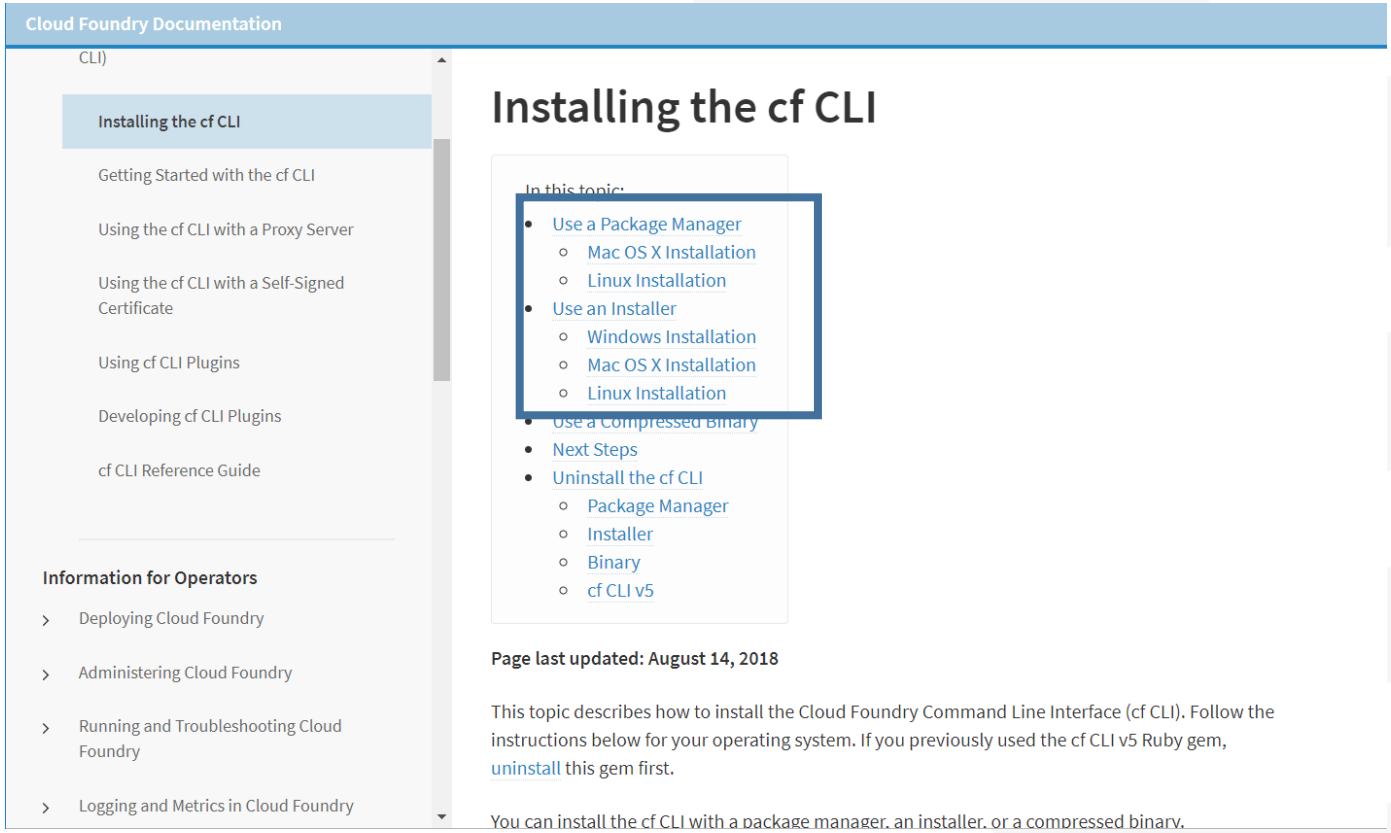
```
---
applications:
- name: rest-api-one
  memory: 1G
  host: rest-api-one
  path: ./target/RestApiOne-0.0.1-SNAPSHOT.jar
  buildpack: java_buildpack
env:
  JAVA_OPTS: -Djava.security.egd=file:///dev/urandom
```

4. 배포 – CLI 로 배포하기

배포 방법은 CLI, IDE plugin등이 있습니다.

CLI Install

https://docs.cloudfoundry.org/cf-cli/install-go-cli.html



CLI login – cf push

배포하려는 소스코드의 디렉토리로 이동

```
infranics@hans:/mnt/d/test/paasta_rest_app/RestUI$ cf login
API endpoint: https://api.paas-ta.co.kr

Email> testuser

Password>
Authenticating...
OK

Targeted org testuser

Targeted space dev



API endpoint:   https://api.paas-ta.co.kr (API version: 2.94.0)
User:           testuser
Org:            testuser
Space:          dev
infranics@hans:/mnt/d/test/paasta_rest_app/RestUI$
```

4. 배포 - CLI 로 배포하기

CLI 배포 – cf push

```
infranics@hans:/mnt/d/test/paasta_rest_app/RestUI$ cf push
Pushing from manifest to org testuser / space dev as testuser...
Using manifest file /mnt/d/test/paasta_rest_app/RestUI/manifest.yml


Deprecation warning: Specifying app manifest attributes at the top level is deprecated. Found: env.
Please see http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html#deprecated for alternati


Using manifest file /mnt/d/test/paasta_rest_app/RestUI/manifest.yml

Updating app rest-ui-test in org testuser / space dev as testuser...
OK

Using route rest-ui-test.paas-ta.co.kr
Uploading rest-ui-test...
Uploading app files from: /tmp/unzipped-app359424878
```

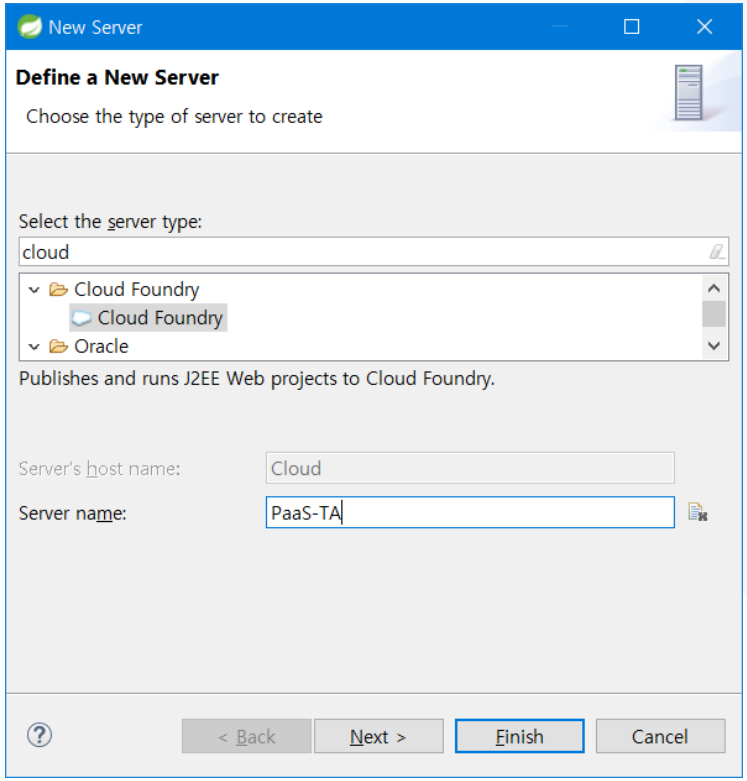
```
Showing health and status for app rest-ui-test in org testuser / space dev as testuser...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: rest-ui-test.paas-ta.co.kr
last uploaded: Mon Nov 12 09:06:16 UTC 2018
stack: cflinuxfs2
buildpack: java_buildpack

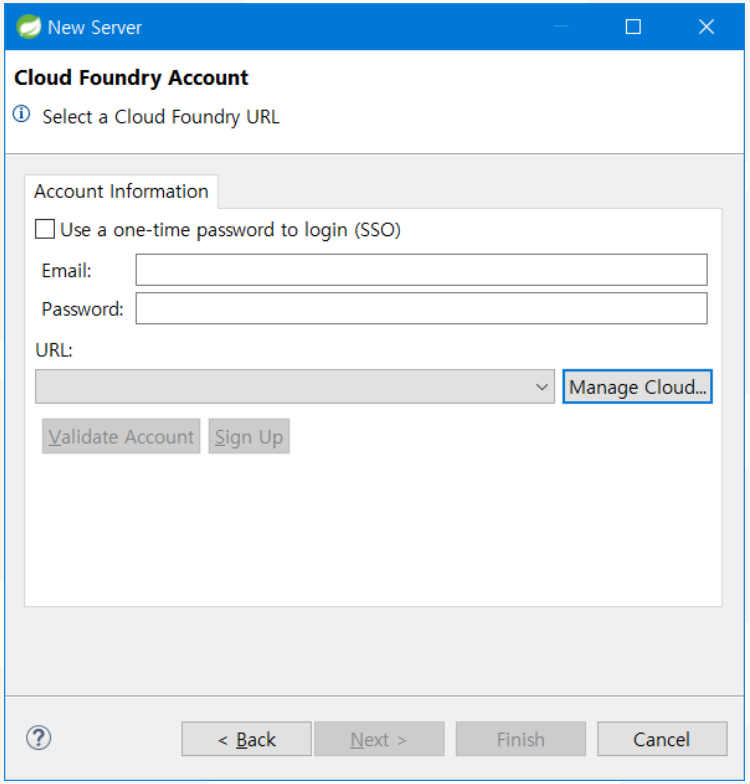
     state     since                  cpu    memory      disk      details
#0   running   2018-11-12 06:06:54 PM   0.0%   0 of 512M   0 of 1G
infranics@hans:/mnt/d/test/paasta_rest_app/RestUI$
```

설정

Server > New > New Server
Cloud Foundry 선택

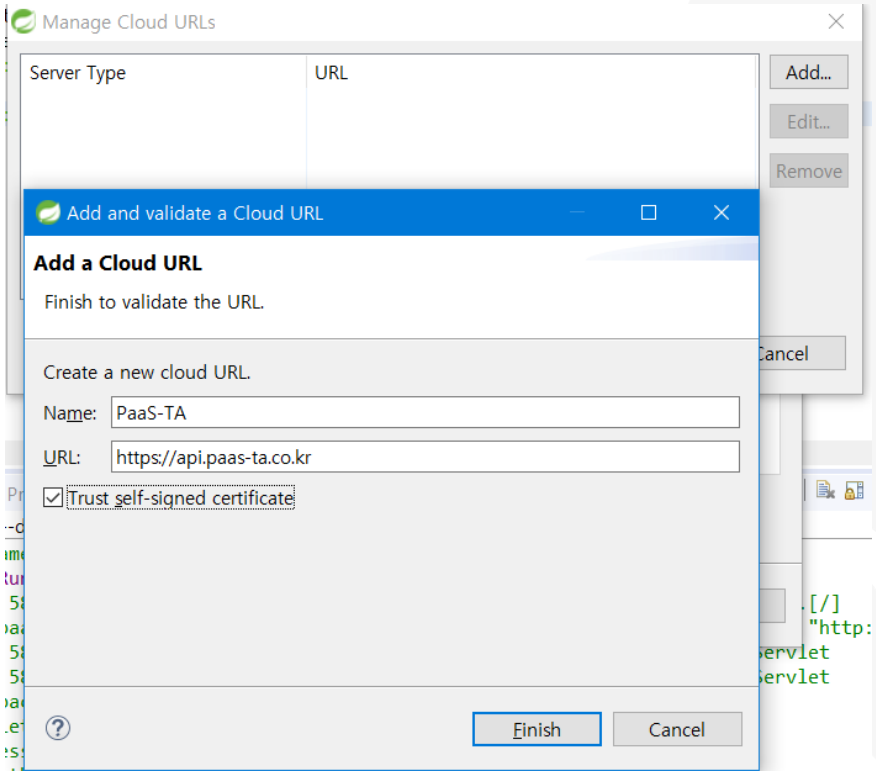


Manage Cloud… 선택 하여
접속할 endpoint API설정

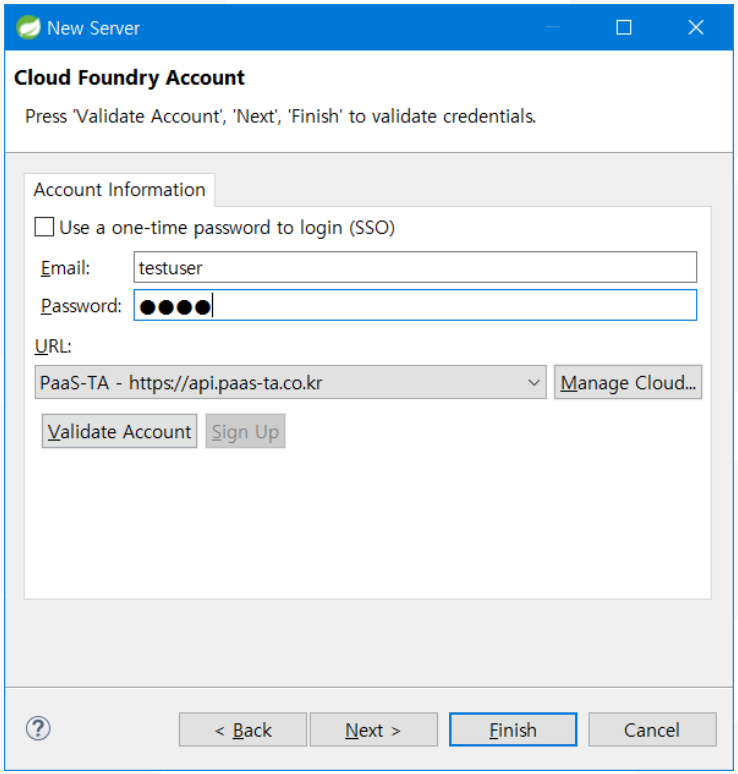


설정

Endpoint API 설정



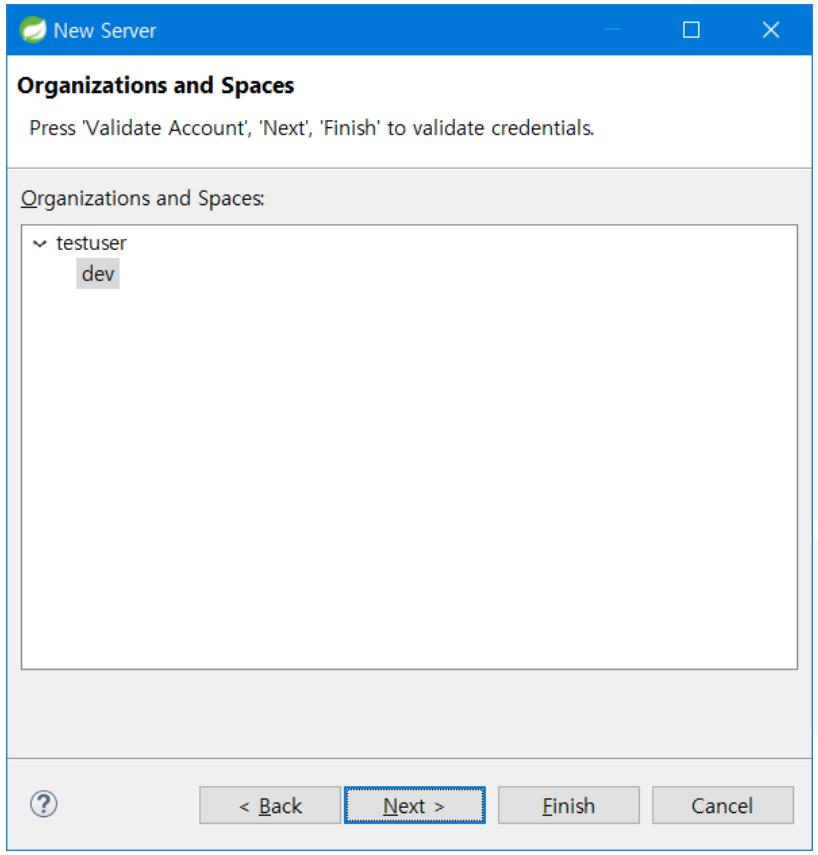
계정정보 입력 및 API선택 후 Validate Account 클릭



설정

API접속 이후 org, space확인 가능

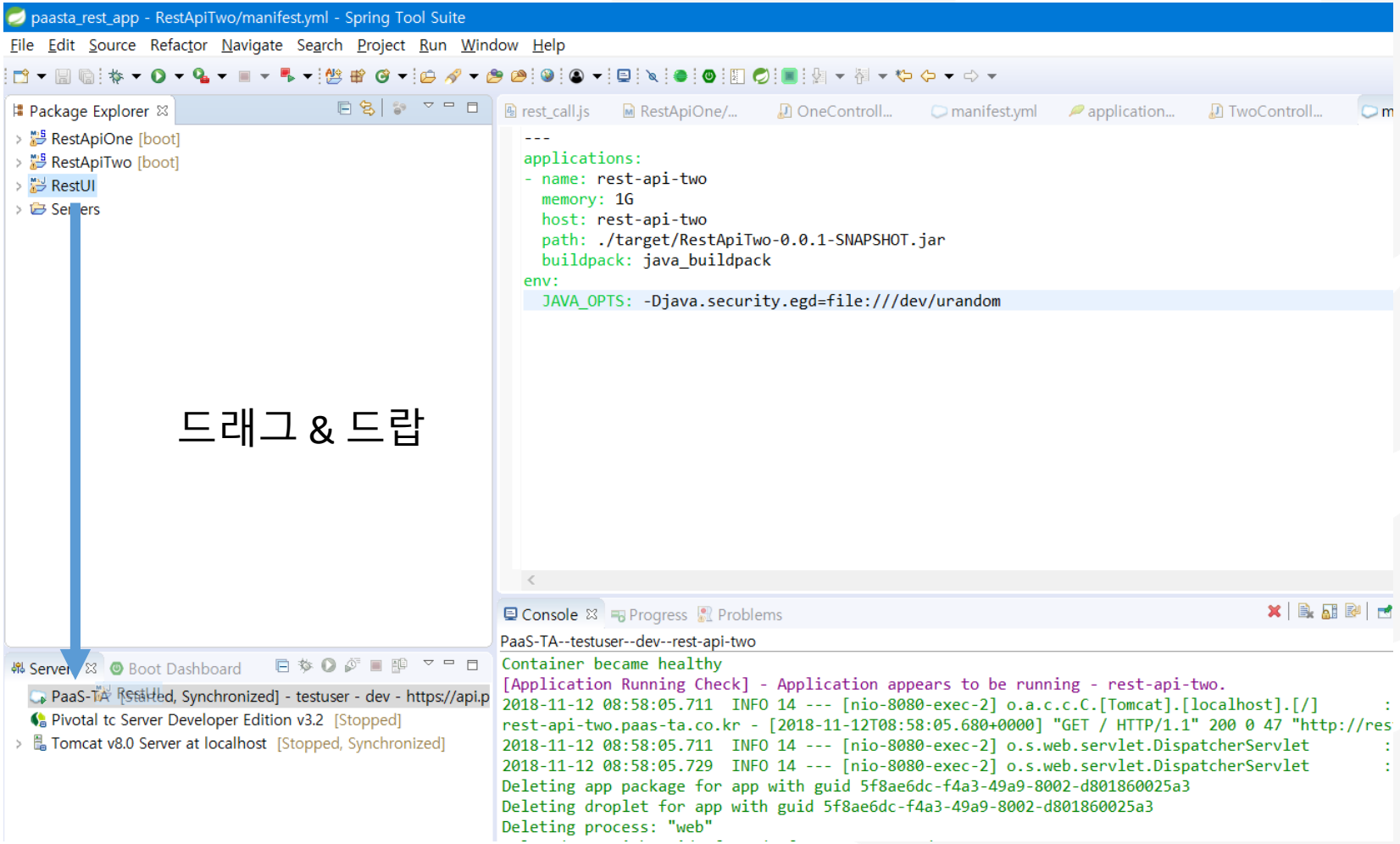
Next > Finish



4. 배포 - IDE 로 배포하기

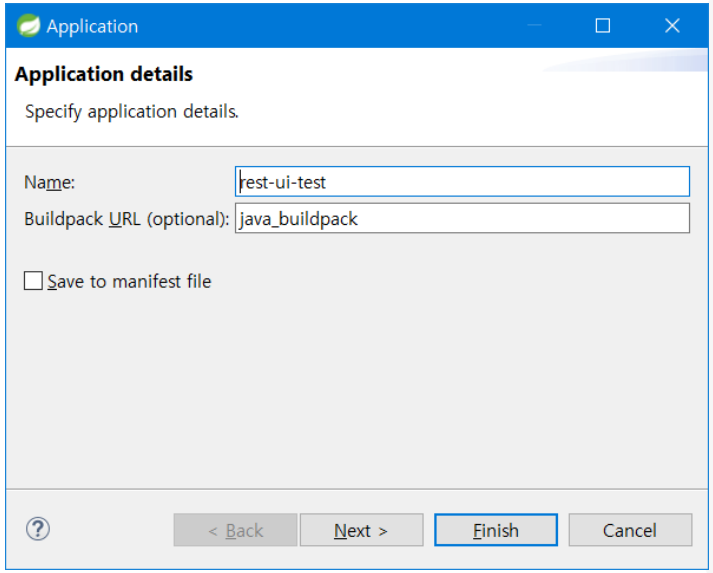
설정

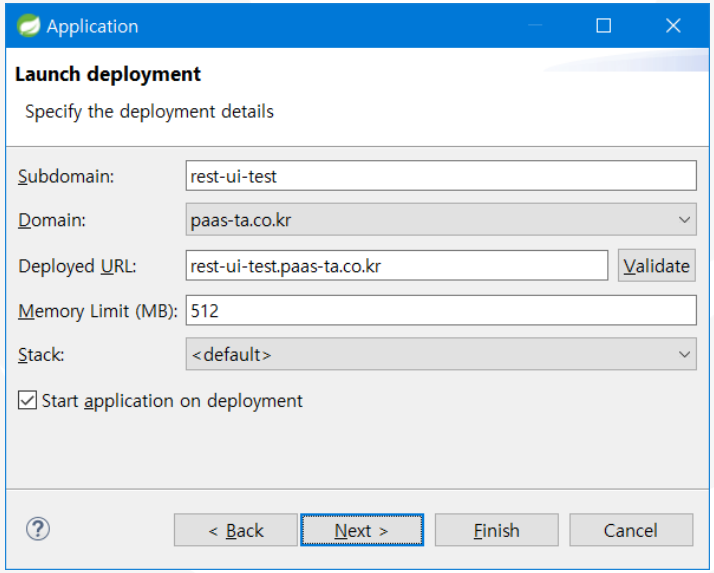
배포하고자 하는 프로젝트를 PaaS-TA에 드래그 & 드랍 하여 배포 끝



설정

배포명, buildpack, domain, memory등의 정보를 입력하고 finish!!

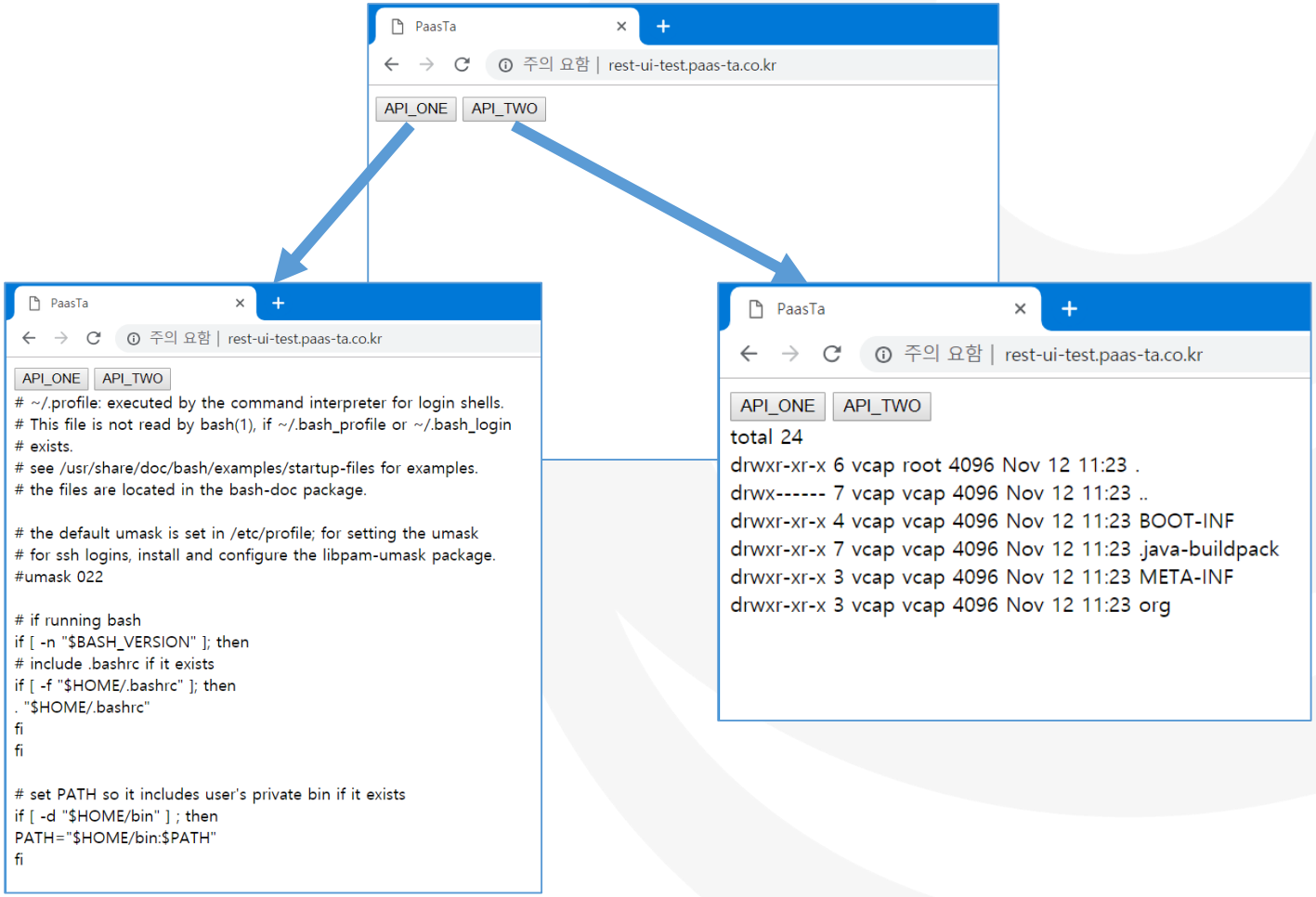




RestApiOne, RestApiTwo 프로젝트도 위와 같은 방법으로 배포

4. 배포 - IDE 로 배포하기

결과 화면



THANK YOU



SysMaster suite
Pro-Active Management

Systeer
Cloud Service

SysCare services
Invisible Guard