Trillo Workbench - Development using IDE

Trillo Workbench (WB) is a runtime to develop applications using model driven and serverless architecture. It means you specify structural parts of an application such as database, domain metadata, etc. as metadata. You add application logic using serverless functions (referred to as Trillo function). Both are deployed on the Trillo Workbench via UI or a git repository. Trillo Workbench using your code (metadata and function) makes the application functionality available.

One of the challenges of writing serverless functions is to debug them. This guide describes how to develop serverless functions for Trillo Workbench using IDE.

How are Trillo functions different from Lambda or Cloud Functions?

Although in concept they are similar, the following are important differentiations.

1. Trillo functions use Trillo Workbench APIs.They don't deal with cloud APIs or database connections, etc. Therefore, they require Trillo Workbench to run.
2. Trillo Workbench deploys them therefore they don't require CI/CD pipeline.

Anatomy of Trillo Function

Anatomy of Trillo Function is repeated here for convenience. It is covered at other places such as Trillo Workbench Developer Guide.

Each Trillo Function has the following structures. Each method corresponds to one endpoint. You can add new methods or remove methods that are not needed.

Trillo Workbench publishes each method as an API using the following convention for its url-path.

ds/function/<functionName>/<methodName>

```
import java.util.Map;
import com.collager.trillo.util.Api;
import com.collager.trillo.util.ServerlessFunction;

public class OrderService extends ServerlessFunction {

  @Api(httpMethod="get")
  public Object getItems(Map<String, Object> parameters) {
    return parameters;
  }

  @Api(httpMethod="post")
  public Object addItems(Map<String, Object> parameters) {
    return parameters;
  }

}
```

Developing Using IDE

THe following sections describe concepts and steps to start developing Trillo functions using IDE.

Prerequisites

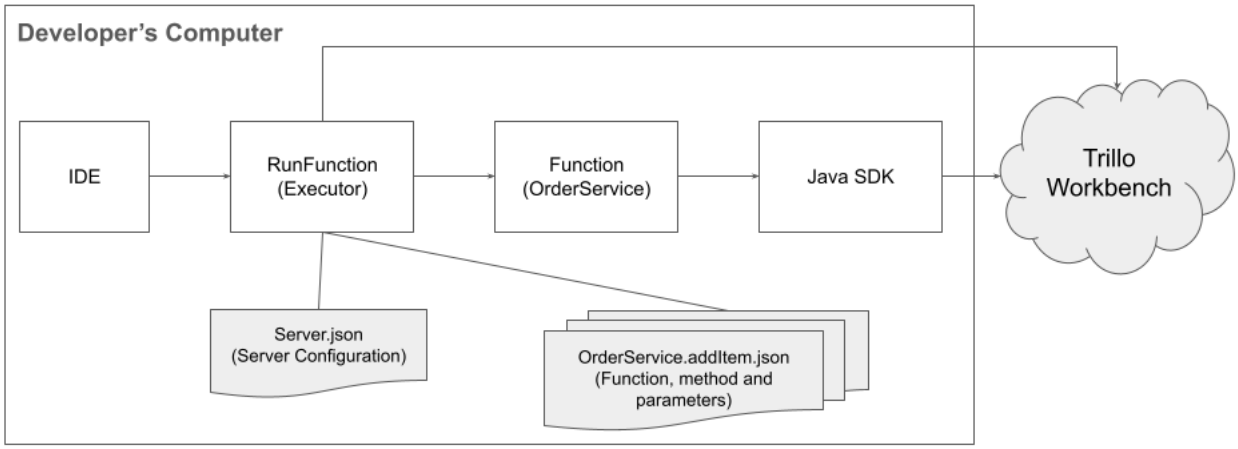
1. Java IDE should be available on your machine. It should be using Java 8 or above.
2. You should have cloned this repository.
3. You should have access to a Trillo Workbench running on cloud (URL, login credentials to its UI).
4. You can use the same credentials to access Trillo Workbench from your client.

How do Trillo functions run inside an IDE?

In an IDE the Trillo function runs through an executor program (RunFunction.java, provided in the repo).

1. The executor reads Workbench configuration from a file (config/Server.json).
2. It reads credentials from environment variables.
    a. TRILLO_WB_USER_ID
    b. TRILLO_WB_USER_PASSWORD

3. Using credentials and serverUrl (see config/Server.json), it connects with the Trillo Workbench , authenticates and acquires an access-token for use in API calls.
4. config/Server.json points to a file containing function details.
5. The function details file provides function name, method name and parameters to invoke it. Using these, the executor invokes the function.



Steps to Develop Trillo functions using IDE

1. New a Java file using IDE and copy the code below, change the class name, method name. Change HTTP method type (get, post, put, delete) in the annotation. **Make sure the function inherits from the ServerlessFunction.**

   Copy the following code. Replace `{{placeHolderName}}` **with your function name and** `postMethodChangeMe` **with your method name.**

```
import java.util.Map;
import com.collager.trillo.util.Api;
import com.collager.trillo.util.ServerlessFunction;
public class {{placeHolderName}} extends ServerlessFunction {
  @Api(httpMethod="post")
  public Object postMethodChangeMe(Map<String, Object> parameters) {
    return parameters;
  }

}
```

2. Write your code using Trillo Workbench APIs. See the references below for more info.

3. Update Trillo Workbench configuration in config/Server.json. (see section below for more information).
4. Specify the function detail file. (see below)
5. Using RunFunction as the main class, start running or debugging.

Configuring Trillo Workbench

Refer to the following sample file for specifying the configuration of Trillo Workbench you plan to use for the development:

```
{
   "serverUrl":"<url of your workbench instance>",
   "userId":"Defined as environemnt varaible TRILLO_WB_USER_ID.",
   "password":"defined as environemnt varaible TRILLO_WB_USER_PASSWORD",

   "functionDetailsFile" : "config/functions/OrderService.getItems.json"
}
```

The following table describes attributes of the configuration file.

Attribute Name	**Description**
serverUrl	Trillo Workbench URL
userId	User id to authenticate with the workbench. This has to be configured as an environment variable **TRILLO_WB_USER_ID.** It is for security reasons. The executor updates configuration at runtime from the environment.
password	Password to authenticate with the workbench. This has to be configured as an environment variable **TRILLO_WB_USER_ID.**
functionDetailsFile	Points to the file containing the details of the function to be executed.

Specifying Functions Details

Refer to the following sample file (**config/functions/OrderService.gerItems.json**) for specifying details of the function and its method to be tested. This file is named as <function name>.<method name>.json. It is so for clarity. But you can name it anything you want.

```
{
  "functionName" : "OrderService",
  "methodName" : "getItems",
  "parameters" : {
      "orderId" : 123
  }
}
```

The following table describes attributes of the function details file.

Attribute Name	**Description**
functionName	This is java class name you are testing / debugging.
methodName	THe particular method you want to test / debug.
parameters	Parameters to be passed to the method.