



realMethods

**Developer
Step-By-Step Guide**

Steps To Take

Step 1 – Decide How To Run

Step 2 – Create a Repository

Step 3 – Choose a Tech Stack

Step 4 – Provide a Business Model

Step 5 – Modify Sample Config Files

Step 6 – Generate The Application

Steps 1 (a) – Decide How To Run

a.) Get the realMethods Command Line Interface (CLI) *

```
npm install -g realmethods-cli
```

We will assume
using the CLI

b.) Use the realMethods CircleCI Orb

```
https://github.com/realmethods-public/orb
```

c.) Clone the sample GitLab project

```
https://gitlab.com/realmethods-public/gitlab-appgen
```

d.) Clone the sample Bitbucket project

```
https://bitbucket.org/realMethods-public/bitbucket-appgen/src/master/
```

Steps 1 (b) – Decide How To Run (CLI Option)

Visit platform.realmethods.com and register as a new user. Copy the API token and use it to initialize the Command Line Interface

```
realmethods_cli init b4cbIZ1STkqHBxok
```

Registration

User Id	Password	Match
<input type="text" value="netuser"/>	<input type="password" value="*****"/>	<input type="password" value="*****"/>
First Name:	Last Name	
<input type="text" value="John"/>	<input type="text" value="Doe"/>	
Email	Company	
<input type="text" value="john.doe@gmail.com"/>	<input type="text" value="ACME Software"/>	

Click to accept the End User License Agreement [Read EULA](#)

Done registering and your API token is **b4cbIZ1STkqHBxok**.

[Register New Account](#) [Close](#)


Steps 2 – Create a Git Repository

Choose the Git platform of your choice, but we will assume GitHub for this guide

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner



 complexmathguy ▾

Repository name *

Django ✓

Great repository names are short and memorable. Need inspiration? How about [automatic-chainsaw?](#)

Description (optional)

-  **Public**
Anyone can see this repository. You choose who can commit.
-  **Private**
You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

Steps 3 – Choose a Tech Stack

Use the CLI to display the list of available technology stacks

```
realmethods_cli stack_list public --output pretty
```

id	name	version	contributor	scope	type	status
5	ASP.NET Core	1.0	dev@realmethods.com	PUBLIC	webapp	production
6	AWS Lambda MongoDB	1.0	dev@realmethods.com	PUBLIC	serverless	production
7	AWS Lambda	2.0	dev@realmethods.com	PUBLIC	serverless	production
8	Angular7	1.0	dev@realmethods.com	PUBLIC	mobile	production
9	Django	1.0	dev@realmethods.com	PUBLIC	webapp	production
10	Google Functions	1.0	dev@realmethods.com	PUBLIC	serverless	production
11	Spark Micro Web Framework	1.1	dev@realmethods.com	PUBLIC	webapp	production
12	Spring Core	1.1	dev@realmethods.com	PUBLIC	webapp	production
13	Spring MongoDB	1.1	dev@realmethods.com	PUBLIC	restfulapi	production
14	Struts2	1.4	dev@realmethods.com	PUBLIC	webapp	production

*Note: Either the id or the name can be used when specifying the tech stack. For illustration purposes, use **Django**.*

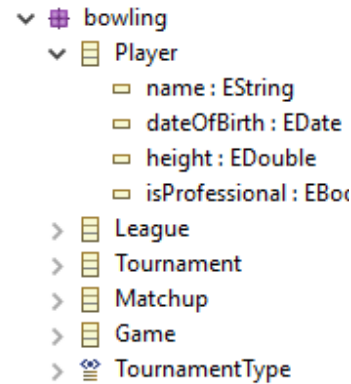
Steps 4 – Provide a Business Model

UML – (<http://argouml.tigris.org//>)



Class Diagram

EMF – <https://www.eclipse.org/>



Exported SQL Script

```
1 CREATE DATABASE /*!32312 IF NOT EXISTS*/ `classic`
2
3 USE `classic`;
4
5 /*Table structure for table `customers` */
6
7 DROP TABLE IF EXISTS `customers`;
8
9
10 CREATE TABLE `customers` (
11   `customerNumber` int(11) NOT NULL,
12   `customerName` varchar(50) NOT NULL,
13   `contactLastName` varchar(50) NOT NULL,
14   `contactFirstName` varchar(50) NOT NULL,
15   `phone` varchar(50) NOT NULL,
16 )
```

JSON – (<https://realmethods.com/home/models/#json>)



Plain Old Java Objects (POJO)







Steps 5(a) – Modify Sample Config Files

Each install method will give access to a samples directory

bitbucket-appgen / samples

Name	Size	Last commit	Message
↑ ..			
📁 git		2019-04-10	updated config files
📁 models		2019-04-10	initial commit
📁 options		2019-04-10	initial commit
📁 yamls		2019-04-10	updated config files

Steps 5(b) – Modify Sample Config Files

-  **git** → YAML file containing credential and server details related to the repository created in Step 2. Locate the Django section and modify the *username, password and repository* fields. We will reference the *github.yml* for illustration purpose.
-  **models** → Sample model files to consider using when first testing things out.
-  **options** → Stack related options. Choose and modify *Django.options.json* file with your specific details. *Pay attention to the nexus section parameters if you are deploying artifacts. Set this section's inUse option to false if not deploying artifacts.*
-  **yamls** → Consider reviewing *generate-django.yml*. If thus far you have only modified what has been suggested, there is no need to modify this file.

Steps 6 – Generate the Application

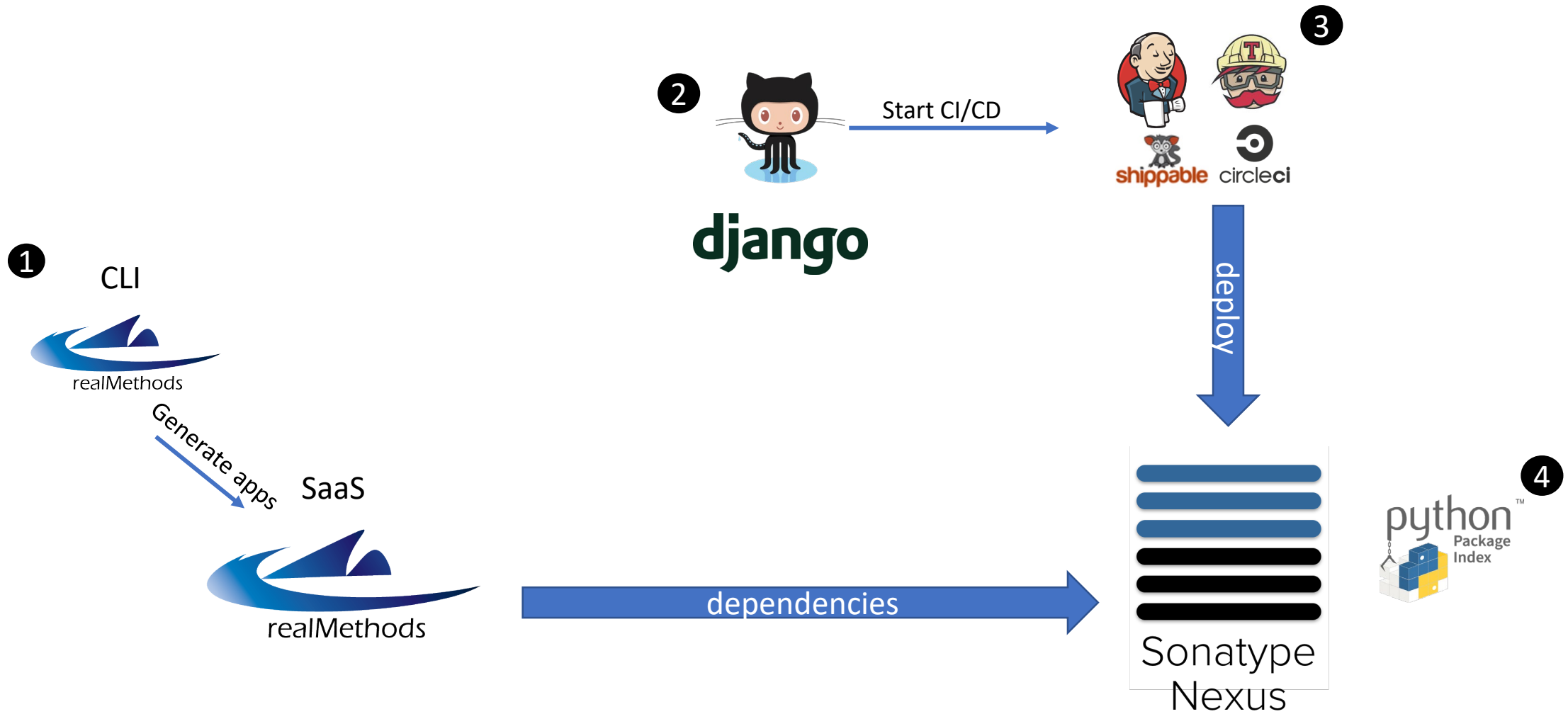
Using the CLI....

```
realmethods_cli app_generate ./samples/yamls/generate-django.yml  
-g ./samples/git/github.yml  
-o ./samples/options/sonatype/Django.options.json  
-m ./path-to-your-model-file
```

If using....

- realMethods CircleCI Orb
- GitLab project
- or
- Bitbucket project

...then refer to the README.md file



1. Use `realmethods_cli` to generate app
2. See generated code checked into repository named Django
3. If the repository is tied to a supported CI/CD platform, a build/test/deploy can occur
4. Observe the python application archive stored in the designated Nexus repository