



devonfw getting started guide

Updated at 2020-01-31 08:48:59 UTC

Table of Contents

I: Getting Started	1
1. Introduction	2
1.1. What is devonfw?	2
1.2. Why should I use devonfw?	5
1.3. devonfw-ide Download and Setup	7
2. Guides	8
2.1. Build Your First devonfw Application	8
2.2. Explore Our devonfw Sample Application	9
3. Further Information	11
3.1. Repository Overview	11
3.2. Links to our Community	12
3.3. Contributing	12
3.4. Code of Conduct	12

Part I: Getting Started

1. Introduction

1.1. What is devonfw?



Welcome to the **devonfw** platform. This is a product of the CSD (Custom Solution Development) industrialization effort to establish a standardized platform for custom software development within Capgemini APPS2. This platform is aimed at engagements, in which clients don't specify the use of a predefined technology stack. In these cases we can offer a proven alternative as a result of our experience as a group.

devonfw is a development platform aiming for the standardization of processes and the boosting of productivity. It provides an architecture blueprint for server and client applications, alongside a set of tools to deliver a fully functional, *out-of-the-box* development environment.



The devonfw *name* is a registered trademark of [Capgemini](#), but the software and documentation included in *devonfw* are fully open source. Please refer to our [OSS Compliance](#) section for more information.

1.1.1. Building Blocks of the Platform



devonfw uses a state-of-the-art, open source, core reference architecture for the server (these days considered a commodity in the IT-industry) and on top of that an ever increasing number of high-value assets, which are developed by Capgemini.

1.1.2. The devonfw Technology Stack

devonfw is fully open source and consists of the following technology stacks:

Back-End Solutions

For server applications, *devonfw* includes the following solutions:

- [devon4j](#): Server implementation based on Java, [Spring](#) and [Spring Boot](#).
- [devon4net](#): Server implementation based on [.NET](#).
- [devon4node](#): Server implementation based on [NestJS](#).

Front-End solutions

For client applications, *devonfw* includes two solutions based on *TypeScript*, *JavaScript*, *C#* and *.NET*:

- **devon4ng**: Frontend implementation based on [Angular](#) and a hybrid mobile implementation based on [Ionic](#).
- **devon4X**: Mobile implementation based on [Xamarin](#).

1.1.3. Custom Tools

devonfw-ide

The `devonfw-ide` is *not* one monolithic program that is installed with a traditional executable; rather it's a collection of scripts which are invoked via command line to automate several, repetitive development tasks. These scripts then interact with other tools, frameworks, and third-party IDEs to streamline the development workflow.

```

..... ///
..... dd                      ffffff
..... ddd                      ffffff
..... ddd                      ff
..... dddddd eeeeeee uuu      uuu  oooo  nnnnnn  ffffffff ww      ww
..... dddddd eeeeeee uuu      uuu  oooooooooo nnnnnnnnnn ffffffff ww      w      ww
..... dddd ddd eeee  eee  uuu  uuu  oooo  oooo nnn  nnn  ff      ww      www      ww
..... ddd ddd eeeeeeeeee uuu  uuu  ooo  ooo nnn  nnn  ff      ww      wwwww      ww
..... ddd ddd eeeeeeeeee uuuuuuu  ooo  ooo nnn  nnn  ff      ww      wwwww      ww
..... ddd ddd eeee      uuuuuu  ooo  ooo nnn  nnn  ff      wwwww      wwwww
..... dddddd eeeeeeeeee uuu      oooooooooo nnn  nnn  ff      ww      ww
..... dddd      eeeeeee      u      oooo      nnn  nnn  ff      w      w
.....

```

Current version of devon-scripts is

usage: devon [command [args]]
Setup devonfw IDE environment and optionally launch commands.

Commands:

- build
- cicdgen
- eclipse
- gradle
- help
- ide
- intellij
- java
- jenkins
- mvn
- ng
- node
- npm
- release
- vscode
- yarn

For further details use:
devon help <command>

The advantage of this approach is, that you can have as many instances of the *devonfw-ide* on your

machine as you need — for different projects with different tools, tool versions and configurations. No need for a physical installation and no tweaking of your operating system required!

Instances of the *devonfw-ide* do not interfere with each other, nor with other installed software. The package size of the *devonfw-ide* is initially very small, the setup is simple, and the included software is portable.

IDEs

It supports the following IDEs:

- [Eclipse](#)
- [Visual Studio Code](#)
- [IntelliJ IDEA](#)

Platforms

It supports the following platforms:

- [Java](#) (see also [devon4j](#))
- [Node.js](#) (see also [devon4node](#))
- [Angular](#) (see also [devon4ng](#))
- [C#](#) (see also [devon4net](#))

Build-Systems

It supports the following build-systems:

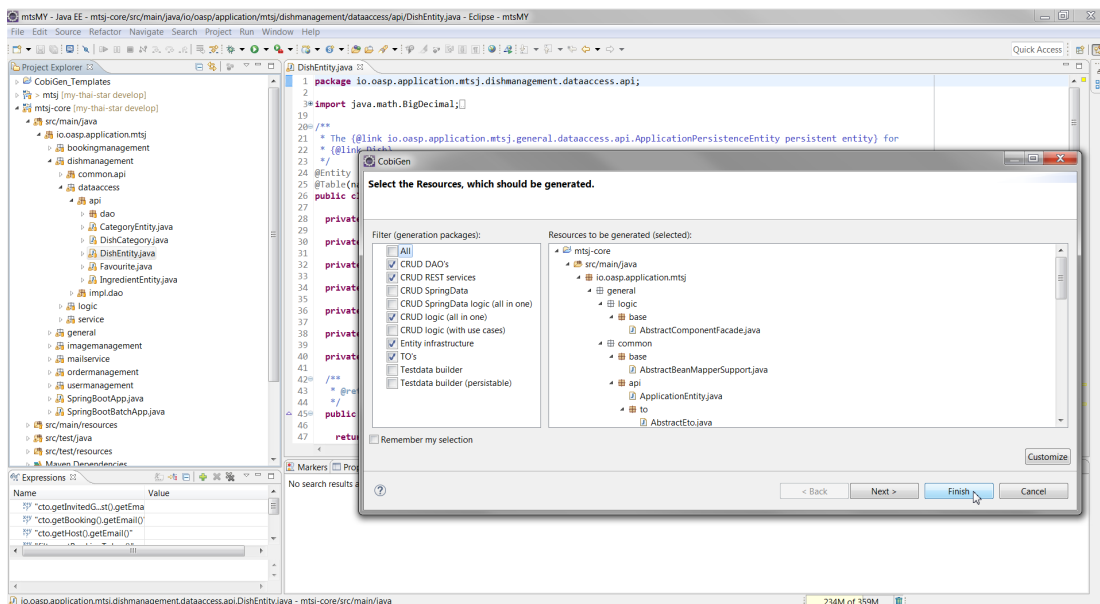
- [Maven](#)
- [NPM](#)
- [Gradle](#)



Other IDEs, platforms, or tools can easily be integrated as [commandlets](#).

CobiGen

[CobiGen](#) is a code generator included in the *devonfw-ide*, that allows users to generate the project structure and large parts of the application component code. This saves a lot of time, which is usually wasted on repetitive engineering tasks and/or writing boilerplate code.



Following the same philosophy as the devonfw-ide, *CobiGen* bundles a new command line interface (CLI), that enables the generation of code using only a few commands. This approach also allows us to decouple CobiGen from *Eclipse* and use it alongside *VS Code* or *IntelliJ IDEA*.

1.2. Why should I use devonfw?

devonfw aims to provide a framework for the development of web applications based on the Java EE programming model. It uses the Spring framework as its Java EE default implementation.

1.2.1. Objectives

Standardization

We don't want to keep reinventing the wheel for thousands of projects, for hundreds of customers, across dozens of countries. For this reason, we aim to rationalize, harmonize and standardize the development assets for software projects and industrialize the software development process.

Industrialization of Innovative Technologies & “Digital”

devonfw's goal is to standardize & industrialize. But this applies not only to large volume, “traditional” custom software development projects. devonfw also aims to offer a standardized platform which contains a range of state-of-the-art methodologies and technology stacks. devonfw supports agile development by small teams utilizing the latest technologies for projects related to Mobile, IoT and the Cloud.

Deliver & Improve Business Value



Efficiency

- Up to 20% reduction in time to market, with faster delivery due to automation and reuse.
- Up to 25% less implementation efforts due to code generation and reuse.
- Flat pyramid and rightshore, ready for junior developers.

Quality

- State-of-the-art architecture and design.
- Lower cost on maintenance and warranty.
- Technical debt reduction by reuse.
- Risk reduction due to continuous improvement of individual assets.
- Standardized, automated quality checks.

Agility

- Focus on business functionality, not on technicalities.
- Shorter release cycles.
- DevOps by design — Infrastructure as Code.
- Continuous Delivery pipeline.
- On- and off-premise flexibility.
- PoCs and prototypes in days not months.

1.2.2. Features

Everything in a Single ZIP

The devonfw distributions is packaged in a ZIP file that includes all the custom tools, software and configurations.

Having all the dependencies self-contained in the distribution's ZIP file, users don't need to install or configure anything. Just extracting the ZIP content is enough to have a fully functional *devonfw*.

devonfw — The Package

The devonfw platform provides:

- Implementation blueprints for a modern cloud-ready server and a choice on JS-Client technologies (either open source Angular or a very rich and impressive solution based on commercial Sencha UI).
- Quality documentation and step-by-step quick start guides.
- Highly integrated and packaged development environment based around Eclipse and Jenkins. You will be ready to start implementing your first customer-specific use case in 2h time.
- Iterative eclipse-based code-generator that understands "Java" and works on higher architectural concepts than Java-classes.
- An example application as a reference implementation.
- Support through a large community + industrialization services (Standard Platform as a Service) available in the iProd service catalog.

1.3. devonfw-ide Download and Setup

Please refer to our [devonfw-ide Setup](#) section.

2. Guides

Our goal is to provide a smooth starting experience to all users of *devonfw*, no matter how experienced they are or what their stakeholder role is. To achieve this, we provide a list of recommended guides here:

For Students and Junior Engineers:

- [Build your first devonfw application.](#)
- [Explore the devonfw sample application.](#)

For Senior Engineers and Architects:

- [Explore the devonfw sample application.](#)

For Team Leaders and Product Ambassadors:

- [License management for OSS projects.](#)

2.1. Build Your First devonfw Application

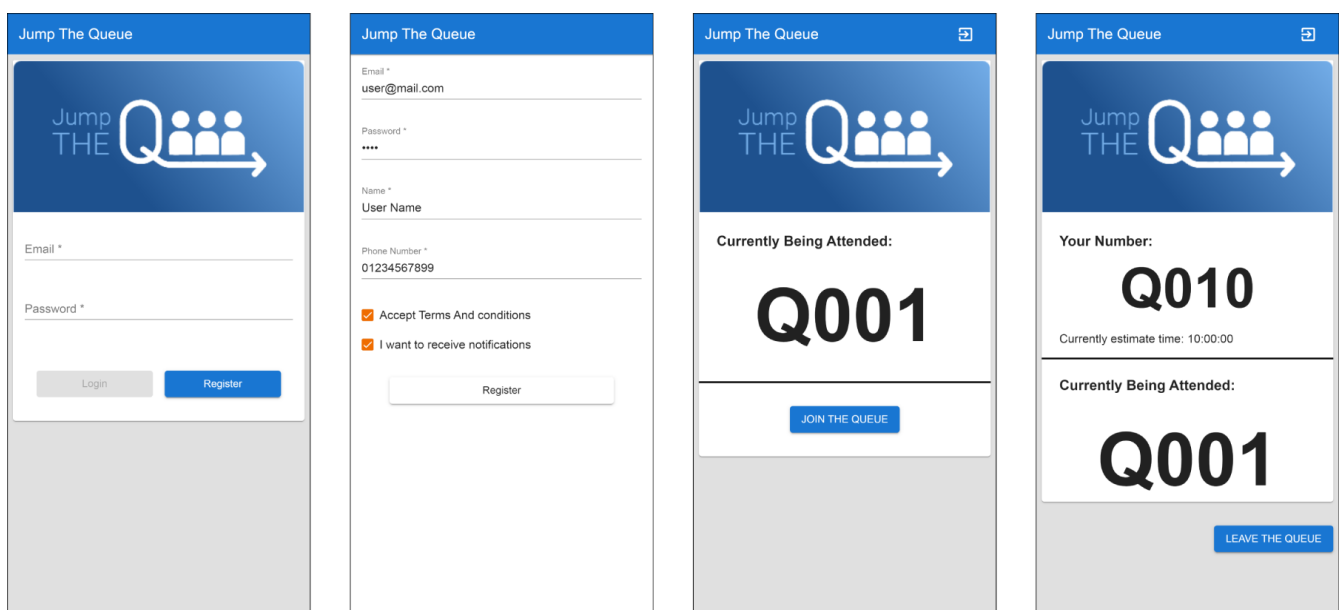
[JumpTheQueue](#) is a small application based on the devonfw framework, which you can create yourself by following our simple step-by-step tutorial. By doing so, you will learn about the app development workflow and gain insight into the design of a professional business information system. Please visit the JumpTheQueue wiki and start working through the tutorial [HERE](#).



The tutorial assumes you have successfully set up the [devonfw-ide](#) previously.

You can also clone the project and explore the finished source code via:

```
git clone https://github.com/devonfw/jump-the-queue.git
```



2.2. Explore Our devonfw Sample Application

MyThaiStar is a complex sample app, that demonstrates the full capabilities of our framework. On this page we will describe how to download and launch the app on your system, so you can test the various functionalities it offers and explore its code.

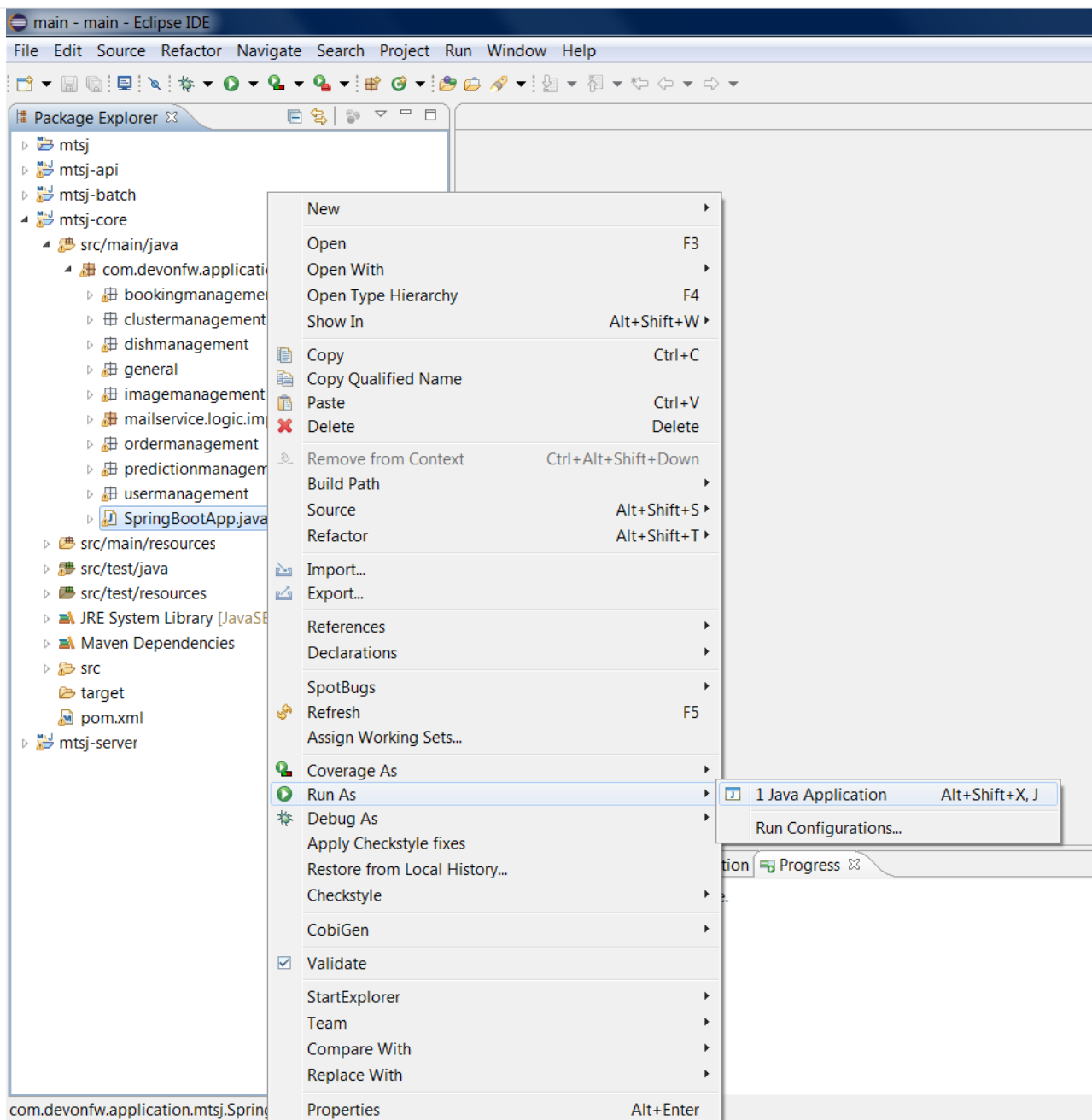


We assume you have successfully set up the [devonfw-ide](#) previously.

1. In the root directory of a *devonfw-ide* directory, right click and select "**Open Devon CMD shell here**" from the Windows Explorer context menu. Then navigate to the main workspace and checkout the MyThaiStar Git repository like this:

```
cd workspaces/main
git clone https://github.com/devonfw/my-thai-star.git
```

2. Perform: `cd my-thai-star`
3. Execute: `devon eclipse ws-up`
4. Execute: `devon eclipse create-script`
5. Go to the root folder of the distribution and run `eclipse-main.bat`
6. In Eclipse navigate to `File > Import > Maven > Existing Maven Projects`, then import the cloned project from your workspace by clicking the "Browse" button and selecting `/workspaces/my-thai-star/java/mts/`.
7. Run the backend by right-clicking `SpringBootApplication.java` and selecting `Run as > Java Application` in the context menu. The backend will start up and create log entries in the Eclipse Console tab.



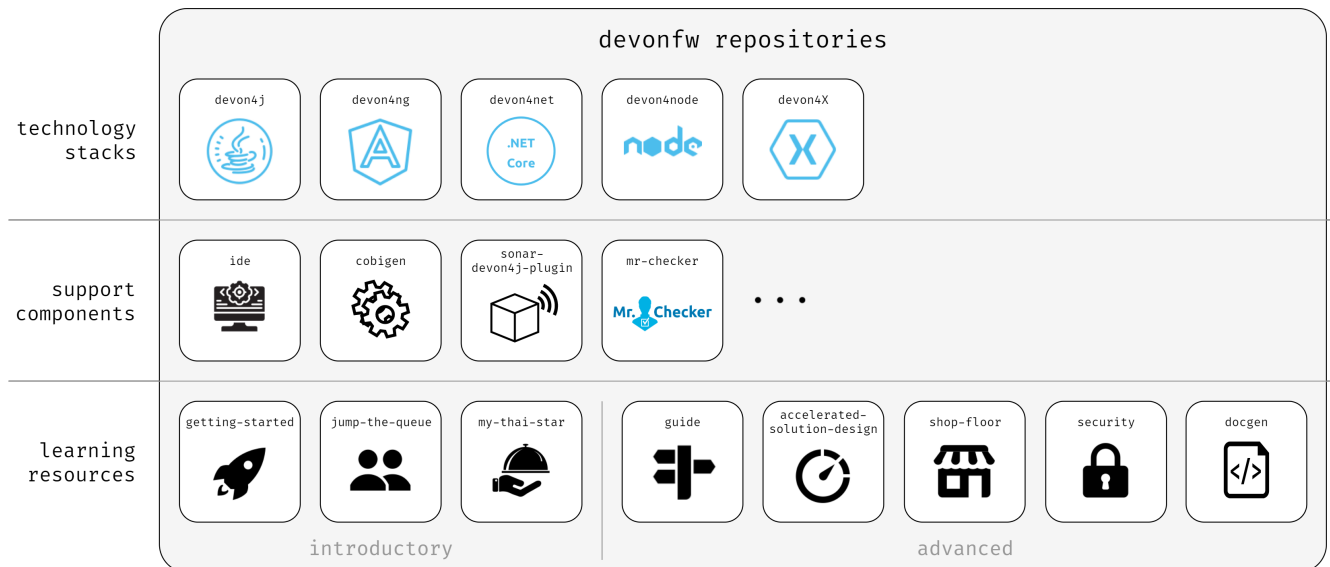
8. Return to your command shell and perform: `cd angular`
9. Execute: `npm install`
10. Execute: `ng serve`
11. Once started, the frontend will be available at localhost:4200/restaurant. Login with the username and password `waiter` and take a look at the various functionalities provided by MyThaiStar.

You should now take a look at both the front- and backend code and familiarize yourself with its structure and concepts, since most devonfw projects follow this exemplary implementation. Please visit the *architecture overview* pages of [devon4ng](#) and [devon4j](#) to learn more about the internal workings of front- and backend.

3. Further Information

3.1. Repository Overview

The GitHub repositories within the [devonfw organization](#) contain the source code and documentation for official devonfw projects.



An overview of the devonfw organization repositories.

The most relevant repositories here are the individual devonfw technology stacks:

- [devon4j](#) (for Java)
- [devon4ng](#) (for Angular)
- [devon4net](#) (for .NET)
- [devon4node](#) (for Node.js)
- [devon4x](#) (for Xamarin)

Our framework also delivers a number of tools and plug-ins that aim to accelerate and streamline the development process, for example:

- [devonfw-ide](#) (for environment/workspace setup)
- [CobiGen](#) (for automated code generation)
- [devon4j-sonar-plugin](#) (for architecture validation)
- [MrChecker](#) (for E2E test automation)

We also provide educational material and reference implementations to aid new users and drive the adoption of our framework, for example:

- [JumpTheQueue](#) (simple, step-by-step app creation tutorial)
- [MyThaiStar](#) (well documented, complex reference application)
- [devonfw-shop-floor](#) (tools and tutorials for CI/CD efforts)

- [accelerated-solution-design](#) (work-methodology for faster app design)
- and many more ...

Projects in early development and prototypes are located in the [devonfw forge](#) repository. They usually remain there until they are ready for broader release or use in production.

3.2. Links to our Community

We strive to foster an active, diverse and dynamic community around devonfw and are relying on modern collaboration tools to do so. Please note that some resources listed here might only be accessible to members or partners of Capgemini.

3.2.1. Microsoft Teams

The devonfw public channel is accessible to everyone who has a Microsoft Teams account. You can find the latest discussions on ongoing development topics here, as well as new commits and pull requests to our repos.

Join us to stay in the loop, and feel free to post your questions regarding devonfw here.

 [devonfw Public Channel](#)

3.2.2. Yammer

Our corporate Yammer channel is accessible to Capgemini employees and members. If you are looking for information or feedback on current and planned projects regarding devonfw, we recommend you ask around here first.

 [devonfw Corporate Yammer](#)

3.2.3. E-Mail

You can reach our dedicated iCSD Support Team via e-mail at:

icsddevonfwsupport.apps2@capgemini.com

3.3. Contributing

Please refer to our [Contributing](#) section.

3.4. Code of Conduct

Please refer to our [Code of Conduct](#) section.