

Angular

With Blueriq 11, we are introducing a new way of front-end development based on the Angular 5+ framework. This will allow you to use today's market standards and build your front-end using a modern technology stack. Furthermore, the usage of Typescript as programming language will help in creating more maintainable code.

The front-end framework consists of two NPM packages, `@blueriq/core` and `@blueriq/angular`. The first package, `@blueriq/core`, is a framework agnostic toolkit that contains interface definitions of the Runtime's UI Rest API and implements common logic for handling API responses. If you plan on using a different framework than Angular, this package should allow you to get started quickly. The second package, `@blueriq/angular`, is an Angular specific package which provides tight integration with the Angular framework, allowing you to write idiomatic Angular components for Blueriq elements.

- [Framework](#)
- [Theming](#)
- [Angular in Production](#)
 - [DNS Routing](#)
 - [Reverse proxy server](#)
 - [Custom themes from development dashboard](#)

Framework

At the core of each Blueriq theme you find the `@blueriq/angular` framework. A detailed guide on what the framework offers and how you may use it is published in the user documentation, of which you can find the latest version [from the API documentation page](#).

Theming

Out of the box, a default theme is distributed with the Blueriq Runtime for showcase purposes. This theme is built-in and can neither be modified nor extended, which ensures that any custom front-end components do not depend on the specific theme version that is supplied with a Blueriq Runtime version. This approach allows for smaller migrations of the Blueriq Runtime, as any theming related changes do not directly affect your custom theme or components.

For you to get started using theme-development, the default theme's source code [is available on Github](#) such that you may fork it and modify/extend it to suit your needs. While being able to keep up-to-date with upstream activity. This theme uses the Angular CLI for development, testing and building your application.

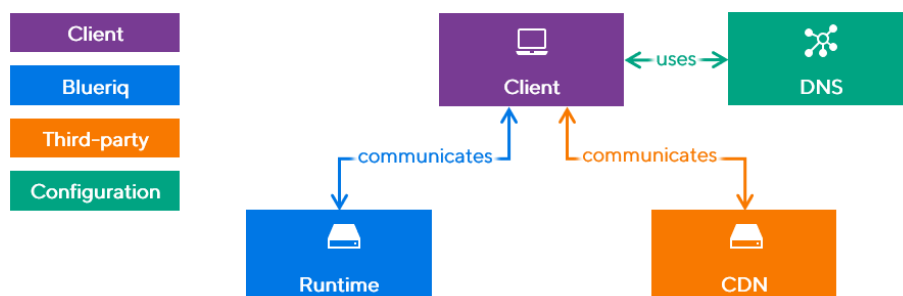
Angular in Production

For bringing your custom Angular theme into production, please refer to the [Angular CLI documentation](#) on building an application into a small set of Javascript bundles for production usage. We recommend serving these files from a different server than the Runtime environment itself, for example a [CDN environment](#) specifically for hosting all static assets.

In order to adhere to the [same-origin security policy](#) without needing to configure additional [CORS headers](#) to allow certain cross-origin requests, we recommend that you either 1) configure a reverse proxy server or 2) configure a subdomain with custom DNS bindings.

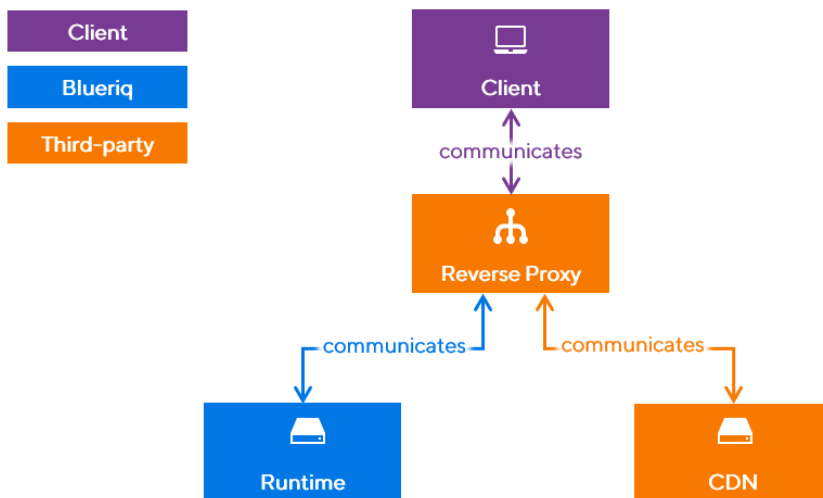
DNS Routing

As an alternative to setting up a reverse proxy server, you may choose to create a subdomain for CDN purposes. In the DNS configuration of your host, you may configure a different IP address for this subdomain, such that asset requests are directly routed to the CDN and API requests arrive at the Runtime instance.



Reverse proxy server

In a reverse proxy server, you may configure redirects based on incoming request paths such that Runtime API requests are redirected to the Runtime backend(s) and static resources are downloaded from the CDN server.



If you are using a clustered Runtime setup, you may choose to implement these redirects in your load balancer layer.



Note that Blueriq ships a prepackaged Material Theme with the Runtime. This theme is only meant to run applications from the development dashboard. Running this theme (or any custom theme that is added as a jar to the Runtime) in production is *not* supported. For production, you should deploy the theme as described above.

Custom themes from development dashboard

Starting Blueriq 14.6 it is possible to start applications using an non prepackaged theme from the development dashboard, more information can be found [here](#).