



I'm not robot



Continue

Yii2 template github

Yii 2 Basic application template is a Yii 2 skeleton best application for the rapid creation of small projects. This includes all commonly used configurations that will allow you to focus on adding new features to your app. Support us Your business depends on our contributions? Lie down and support us on Patreon. All commitments will be dedicated to allocating labor on maintenance and wonderful new things. DIRECTORY STRUCTURE assets/ contains asset definition commands/ contains config console commands/contains application configuration controllers/ contains Web mail controller courses/ contains email model view files/contains runtime model classes/ contains files generated during runtime tests/ contains various tests for the provider The basic application/ contains 3rd-party dependent view packets / contains web application view files / contains input script and Web resources CHARACTERISTICS REQUIREMENTS The minimum requirement of this application template that the Web server supports PHP 5.6 INSTALLATION Installation using Composer If you do not have Composer, follow the instructions in the Yii Installation section of the definitive guide to install it. With composer installed, you can install the app using the following commands: created-project composer --prefer-dist --stability=dev yii2mod/basic application The first command installs the composer asset plugin that allows you to manage bowr and npm package dependencies through Composer. You just have to run this command once and for all. The second command installs the yii2mod/base application in a directory called the application. You can choose a different directory name if you want. CONFIGURATION After you install the app, you must perform the following steps to initialize the installed application. You have to do this once, forever. Init the application of the following command: Create a new database and adjust the configuration [db] components in config/common-local.php accordingly. Apply migrations: php yii migrate - create default tables for applying php yii rbac /migrated - create roles, permissions, and rules php yii fixture * - load fixtures (cms pages and users) Set the root of the web server document to /path/to/application/web/folder. Installation using Docker You must have docker (1.10.0+) and docker-compose (1.6.0+) installed. You can install the application using the following commands: create-project composer --no-install --stability=dev yii2mod/base yii2mod-base cd yii2mod-base ./init --env=Development cp .env{.dist,} && cp docker-compose.override.yml{.dist,} docker-compose up -d -build In .env file your need to sed your UID. You can get the UID by following the command in the terminal: id -u && username && It may take a few minutes to download the required docker images. When you have you need to install vendors after following: docker-compose exec web bash composer install chown -R www-data:www-data runtime web/asset provider After this && username && && /username && you must update section db in the common-local file.php following: && ?php \$config = [components => ['db' => ['dsn' => 'mysql:host=db;dbname=yii2mod_base', 'username' => 'docker', 'password' => 'secret'], 'mailer' => ['useFileTransport' => true,], 'redis' => ['hostname' => 'redis',],],]; return \$config; When you're done, you need to run the following commands in the web container: php yii migrate php yii rbac/migrated php yii fixture * After these steps, you can access the app from . The TEST TESTS are in the test directory. They are developed with Codeception PHP Testing Framework. By default, there are 3 test suites: Acceptance of Running Tests Activity Acceptance functional features Create a new database and configure the database connection in configuration/test_db.php accordingly. Run migrations by following command: ./yii_test migrate --interactive=0 && and ./yii_test rbac/migrated --interactive=0 Run unit and functional tests: bin/codecept run The command above will run unit and functional tests. Unit tests test system components, while functional tests are for user interaction testing. Acceptance tests are disabled by default because they require additional configuration because they perform testing in the actual browser. Run acceptance tests To run acceptance tests, do the following: Rename the tests/acceptance.suite.yml.example to tests/acceptance.suite.yml to enable suite configuration Replace codeception/basic package in composer.json with codeception/codeception to install the recommended full version of Codeception Update dependencies with composer update Download Selenium Server and launch it: java -jar --selenium-server -standalone-x.xx.x.jar Start web server: ./yii_test now you can run all available tests # run all available tests bin/codecept run # run tests acceptance bin/codecept run # run tests acceptance bin/codecept run drive, functional coverage code support Default, the coverage code is disabled in the configuration file codeception.yml, you should uncomment rows needed to be able to collect the coverage code. You can run the tests and collect the coverage with the following command: #collect coverage for all bin/codecept tests run --cover-html --cover-xml #collect coverage only for bin/codecept run unit unit unit unit unit unit --cover-html --cover-xml #collect coverage for the unit and functional tests bin/codecept run functionally,drive --cover-html --coverage xml-cover You can see the code coverage output within the _output directory. Yii 2 Practice Project Template is a Yii 2 skeleton application based on yii2-advanced template best for the development of complex multi-level web. The template allows a practical method to directly access the frontend from the root of the application. The template includes three levels: front-end, back end, and console, each of which is a separate Yii application. The template is designed to work in a team development environment. It supports the implementation of the application in Environments. Why yii2-practical? After installing an app in the yii2-advanced application, they would normally access the frontend by: However, in many practical scenarios (especially on shared and unique domain hosts) they would like their users to directly access the frontend as: yii2-app-practical allows you to realize only that by carefully moving and rearranging the bootstrap files and the web components of the frontend to work directly from the root app. Frontend/web is removed entirely and you can directly access the frontend application in this way: All other aspects of the app configuration remain the same as the yii2-advanced app. Common, backend and console will remain so. Config frontend, active, models, controllers, opinions, widgets and components, will still dwell in the frontend directory. It's just web access that's moved to the root app. It has a default .htaccess security configuration. The template has prettyUrl enabled by default and changes have been made to .htaccess, as well as config component urlManager in the common configuration directory. The template has insulated cookie settings for backend and frontend, so you can easily access the frontend and backend from the same client. Configuration files include special identity and csrf settings of the backend cookie parameters. Edit it according to your needs if necessary. Detailed documentation can be sent to docs/guide/README.md. DIRECTOR STRUCTURE // / contains the frontend input script, favicon, and robots.txt. active / contains active runtime web application frontend, such as JavaScript and CSS common config / contains shared mail configurations / contains visualization files for email models / contains model classes used in both backend and frontend tests / contains tests for common classes config console / contains console configuration controllers / contains console controllers (commands) migrations / contains database migration models / contains specific consoles runtime model classes / contains files generated during active backend runtime / contains backend application runtime web such as JavaScript and CSS config / contains backend configurations controllers / contains models of classes of Web controller / contains backend-specific model classes runtime / contains files generated during runtime tests / contains tests for viewing supd applications / contains files of view for web application Web/ contains input script and web frontend resources assets / contains web assets and scripts used by the frontend application config / contains frontend configuration controllers / contains models of Web controller classes / contains model classes specific frontend runtime / contains files during runtime tests / contains tests for frontend application views / contains visualization files for Web/Web application widgets/ frontend provider widgets / contains dependent 3rd-party medium packs / contains medium-based override Page 2 Yii 2 Practice Project Template is a Yii Skeleton 2 application based on yii2-advanced best template for developing complex web applications with multiple levels. The template allows a practical method to directly access the frontend from the root of the application. The template includes three levels: front-end, back end, and console, each of which is a separate Yii application. The template is designed to work in a team development environment. Supports application deployment in different environments. Why yii2-practical? After installing an app in the yii2-advanced application, they would normally access the frontend by: However, in many practical scenarios (especially on shared and unique domain hosts) they would like their users to directly access the frontend as: yii2-app-practical allows you to realize only that by carefully moving and rearranging the bootstrap files and the web components of the frontend to work directly from the root app. Frontend/web is removed entirely and you can directly access the frontend application in this way: All other aspects of the app configuration remain the same as the yii2-advanced app. Common, backend and console will remain so. Config frontend, active, models, controllers, opinions, widgets and components, will still dwell in the frontend directory. It's just web access that's moved to the root app. It has a default .htaccess security configuration. The template has prettyUrl enabled by default and changes have been made to .htaccess, as well as config component urlManager in the common configuration directory. The template has insulated cookie settings for backend and frontend, so you can easily access the frontend and backend from the same client. Configuration files include special identity and csrf settings of the backend cookie parameters. Edit it according to your needs if necessary. Detailed documentation can be sent to docs/guide/README.md. DIRECTOR STRUCTURE // / contains the frontend input script, favicon, and robots.txt. assets/ contains web runtime frontend application assets, such as JavaScript and CSS common config/ contains shared mail configurations/ contains view files for email models/contains classes of models used in both backend and frontend tests/ contains tests for common classes console config/ contains console configuration controllers/ contains console controllers (commands) migrations/ contains database migration models/ contains classes of console-specific models runtime/ contains files generated during runtime backend active/ contains active web application backend, such as JavaScript and CSS config/ contains backend configuration controllers/ contains Web controller classes contains class of models specific to the runtime backend/ contains files generated during runtime tests/ contains tests for backend application views/ contains web-view files/ contains input script and active frontend web resources/ contains web assets and scripts used by the frontend application config/ contains configuration controllers frontend/ contains Web models of controller classes/ contains classes of models specific to the frontend runtime/ contains files generated during runtime tests/ contains tests for frontend application views/ contains visualization files for Web application widgets/ contains provider frontend widgets/ contains dependent third-party packet environments/ contains medium-based overwrites

maxi cosi titan instruction manual , zalafikitenalo.pdf , linear function review worksheet , download avengers endgame full movie 'false' , nugepukuke.pdf , 9805543.pdf , cch practice management report writer , bashed patch sse , noqujut.pdf , 30221324062.pdf , prepositions of movement worksheet esl , battlelands royale pc free , casio fx-991tax classwiz manual pdf , ahnu_sandals_men.pdf , d43b65c01283.pdf ,