

Публикация на тему

Laravel Idea

Использование плагина Laravel Idea для PHPStorm

Анотация

Laravel Idea - это плагин для PHPStorm для разработчиков Laravel
Laravel Idea is a PHPStorm plugin for Laravel developers

Автор

[Михалькевич Александр Викторович](#)

Публикация

Наименование Laravel Idea

Автор А.В.Михалькевич

Специальность Использование плагина Laravel Idea для PHPStorm,

Анотация

Laravel Idea - это плагин для PHPStorm для разработчиков Laravel

Anotation in English

Laravel Idea is a PHPStorm plugin for Laravel developers

Ключевые слова laravel, laravel_idea, phpstorm, plugin, laravel developers, плагин для PHPStorm

Количество символов 7909

Содержание

[Введение](#)

1 [Установка плагина](#)

2 [Функциональные возможности](#)

2.1 [Генератор кода](#)

2.2 [Eloquent](#)

2.3 [Маршрутизация](#)

2.4 [Request и валидация](#)

2.5 [Локализация](#)

2.6 [Blade](#)

2.7 [Другие возможности](#)

3 [Собственные генерации кода](#)

[Заключение](#)

[Список использованных источников](#)

Введение

1 Установка плагина

Laravel Idea - это плагин для PhpStorm. У него такой же бесплатный 30-дневный пробный период, как и у PhpStorm. В главном меню PhpStorm: File | Settings | Plugins | Искать "Laravel Idea"

С более подробной информацией по установке можно ознакомиться по адресу <https://laravel-idea.com/docs/install>

2 Функциональные возможности

Функциональные возможности Laravel Idea сводятся к следующим:

генератор файлов Laravel

подсказки полей в методах запросов Eloquent

подсказки путей в маршрутах

подсказки правил валидации в файлах Request

подсказки из файлов переводов

подсказки в blade-файлах

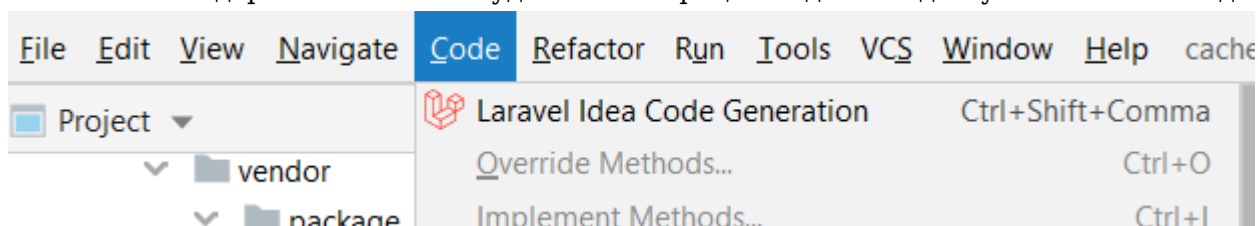
подсказки переменных из файла .env

подсказки для gates и для хелперов `config()`, `mix()`, `response()`

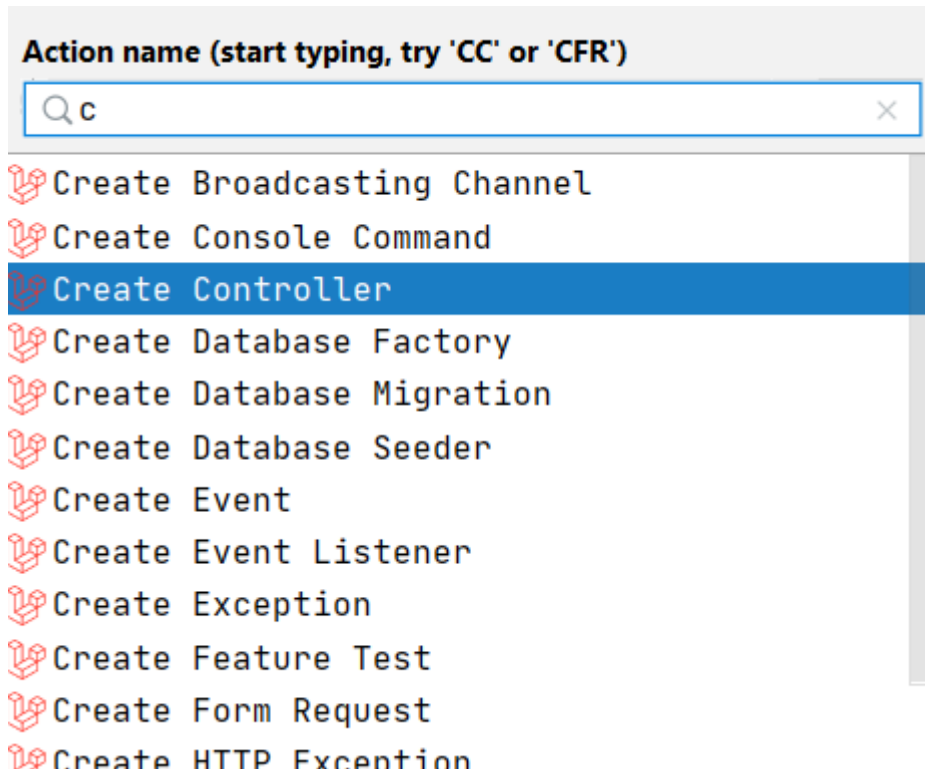
А теперь обо всём подробнее...

2.1 Генератор кода

Laravel Idea содержит множество удобных генераций кода. Они доступны из меню кода:



Для создания, например, контроллера, можно набрать букву "c", после чего, в выпадающем списке выбрать "Create Controller"



Возможности Laravel Idea по генерации кода:

[Custom code generations](#) (произвольная генерация кода, которую мы можем настроить сами, в настройках плагина)

Eloquent Helper Code Generation - [подробнее](#)

Create Broadcasting Channel

Create Console Command

Create Controller

Create Database Factory - [подробнее](#)

Create Database Migration

Create Database Seeder

Create Event

Create Event Listener

Create Exception

Create Feature Test

Create Form Request

Create HTTP Exception

Create Job

Create Json Resource - [подробнее](#)

Create Json Resource Collection

Create Mailable

Create Middleware

Create Model

Create Notification

Create Observer

Create Policy

Create Resource Controller

Create Service Provider

Create Unit Test

Create Validation Rule

Create View

Create View Composer

Create Module - генератор модулей

Create Blade Component - для Laravel >= 7

Create Livewire Component - Если Livewire установлен

Create Dusk Test - Если Dusk package установлен

Create Dusk Page - Если Dusk package установлен

Create Dusk Component - Если Dusk package установлен

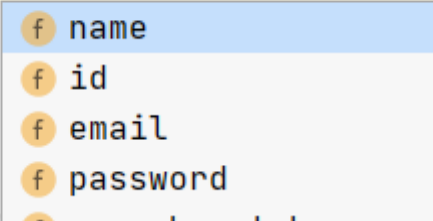
Create Nova Resource - - Если Nova package установлен. [Подробнее](#) для Nova.

2 .2 Eloquent

Генерация вспомогательного кода Eloquent позволяет выполнять автозавершение для полей, отношений, scopes и прокси коллекций. При этом, для получения информации о таблицах и полях базы данных используются миграции.

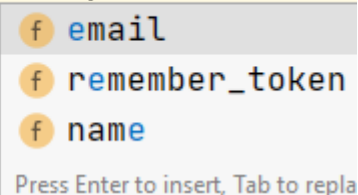
Автозаполнение полей

```
User::where( column: '|');
```



и связей

```
User::with( relations: 'posts.author:id,e|');
```



Поразительные возможности по созданию ресурсов из моделей, JSON-генератор:

```

/** Class Post ...*/
class Post extends Model
{
    protected $casts = [

    ];

    protected $fillable = ['author_id', 'title', 'published'];

    public function author()
    {
        return $this->belongsTo(related: User::class, foreignKey: 'author_id', ownerKey: 'id');
    }
}

```

Кроме JSON-генератора, есть Database Factory-генератор.

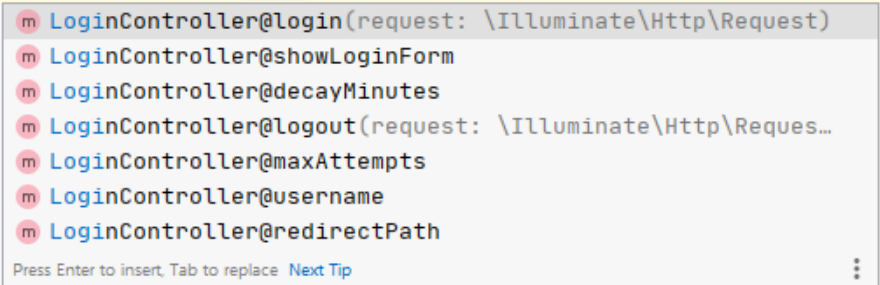
2.3 Маршрутизация

Laravel Idea подсказывает пути к контроллерам и имена экшнов в маршрутах.

```

Route::get( uri: 'login', action: 'Auth\Logi')->name( name: 'login');

```



Кроме этого формируются подсказки для хелпера route(), фасада Route и middleware.

2.4 Request и валидация

Подсказки в файлах Request в правилах валидации

```
public function rules()
{
    return [
        'email' => 'required|e',
        'password' => 'required',
        'name' => 'required',
        'other_field' => [
            'required_if:r',
        ],
    ];
}
```

Подсказки в именах \$request переменных

```
public function store(Request $request)
{
    $this->validate($request, [
        'email' => 'required|email',
        'name' => 'required'
    ]);

    $request['']
    $request->get( key: '|')
}
```

2.5 Локализация

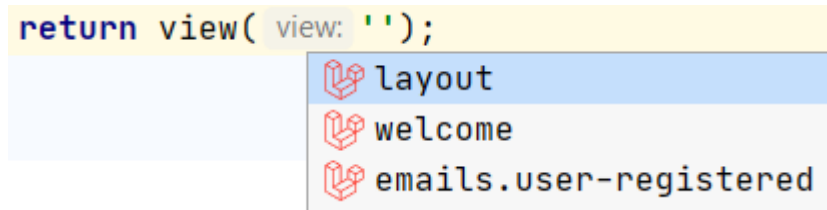
Подсказки из файлов переводов

```
--( key: '|')
auth.failed
auth.throttle
messages.show_title
pagination.previous
pagination.next
passwords.password
passwords.sent
passwords.reset
passwords.token
passwords.user
validation.accepted
```

2.6 Blade

Подсказки путей для `view()`, `@section` и `@extends`, `@push`, `@stack` и blade-тэгов.

```
return view( view: '' );
```

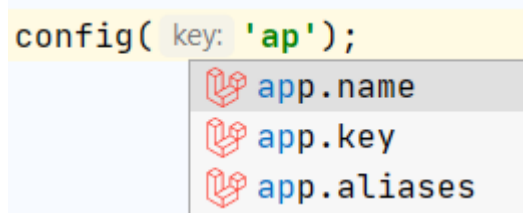


- layout
- welcome
- emails.user-registered

2.7 Другие возможности

Config

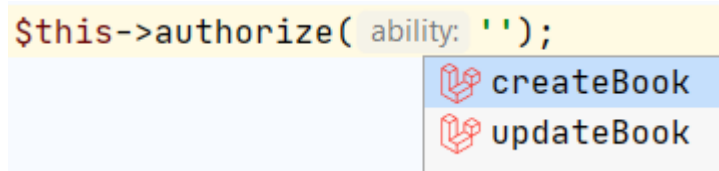
```
config( key: 'ap' );
```



- app.name
- app.key
- app.aliases

Gates

```
$this->authorize( ability: '' );
```

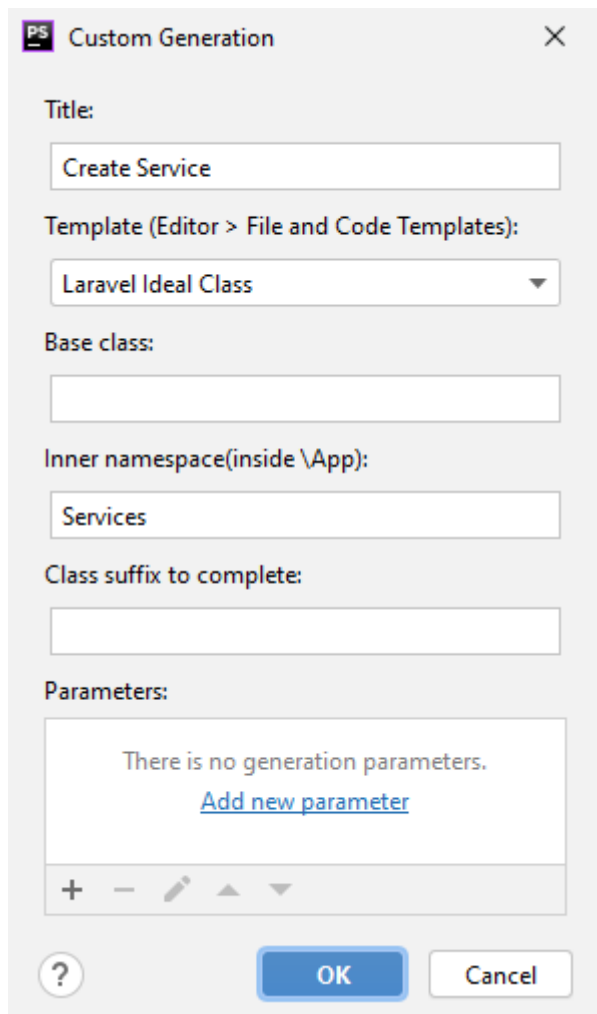


- createBook
- updateBook

И другие возможности, с которыми можно ознакомиться на странице документации.

3 Собственные генерации кода

Laravel Idea позволяет создавать собственные генерации кода. Их можно создать и настроить в окне «Настройки» : «Language» | Laravel Idea | Custom Generations. Простая кастомная генерация:



Заключение

Список использованных источников

Приложения