

Справочник по Yii2

Yii Framework 2.0 API

Класс	Описание
Yii	вспомогательный класс, обслуживающий общие функциональные возможности.
YiiRequirementChecker	позволяет проверить, соответствует ли текущая система требованиям для запуска приложения Yii..
yii\BaseYii	основной вспомогательный класс для структуры Yii.
yii\base\Action	базовый класс для всех классов действий контроллера.
yii\base\ActionEvent	представляет параметр события, используемый для события действия.
yii\base\ActionFilter	базовый класс для фильтров действий.
yii\base\Application	является базовым классом для всех классов приложений.
yii\base\ArrayAccessTrait	предоставляет реализацию для IteratorAggregate , ArrayAccess и Countable .
yii\base\Arrayable	интерфейс, который должен реализовываться классами, которые хотят поддерживать настраиваемое представление своих экземпляров.
yii\base\ArrayableTrait	обеспечивает общую реализацию интерфейса yii\base\Arrayable
yii\base\BaseObject	базовый класс, который реализует свойство. (<i>property</i>).
yii\base\Behavior	является базовым классом для всех классов поведения.
yii\base\BootstrapInterface	интерфейс, который должен быть реализован классами, которые хотят участвовать в процессе начальной загрузки приложения.
yii\base\Component	базовый класс, который реализует свойства, события и поведение. (<i>property</i> , <i>event</i> и <i>behavior</i>).
yii\base\Configurable	интерфейс, который должен реализовываться классами, которые поддерживают настройку своих свойств с помощью последнего параметра его конструктору.
yii\base\Controller	является базовым классом для классов, содержащих логику контроллера.
yii\base\DynamicModel	модельный класс, в основном используемый для поддержки проверки ad hoc.
yii\base\Exception	представляет ошибку PHP.
yii\base\ErrorHandler	обрабатывает неперехваченные ошибки и исключения PHP.
yii\base\Event	является базовым классом для всех классов событий.
yii\base\Exception	представляет собой общее исключение для всех целей.
yii\base\ExitException	представляет собой обычное завершение приложения.
yii\base\InlineAction	представляет собой действие, которое определяется как метод контроллера.
yii\base\InvalidCallException	представляет собой исключение, вызванное неправильным вызовом метода.
yii\base\InvalidConfigException	представляет собой исключение, вызванное неправильной настройкой объекта.
yii\base\InvalidParamException	представляет собой исключение, вызванное неверными

	параметрами, переданными методу.
yii\base\InvalidRouteException	представляет собой исключение, вызванное недействительным маршрутом.
yii\base\InvalidValueException	представляет собой исключение, вызванное функцией, возвращающей значение неожиданного типа.
yii\base\Model	является базовым классом для моделей данных.
yii\base\ModelEvent	представляет параметр, необходимый для событий yii\base\Model
yii\base\Module	является базовым классом для классов модулей и приложений.
yii\base\NotSupportedException	представляет собой исключение, вызванное доступом к функциям, которые не поддерживаются.
yii\base\Object	базовый класс, реализующий свойство. (<i>property</i>)
yii\base\Request	представляет запрос, который обрабатывается с помощью yii\base\Application .
yii\base\Response	представляет собой ответ yii\base\Application на yii\base\Request .
yii\base\Security	предоставляет набор методов для обработки общих задач, связанных с безопасностью.
yii\base\StaticInstanceInterface	интерфейс для предоставления статических экземпляров классам, которые могут использоваться для получения метаданных класса, которая не может быть выражена в статических методах.
yii\base\StaticInstanceTrait	предоставляет методы для удовлетворения интерфейса yii\base\StaticInstanceInterface
yii\base\Theme	представляет собой тему приложения.
yii\base\UnknownClassException	представляет собой исключение, вызванное использованием неизвестного класса.
yii\base\UnknownMethodException	представляет собой исключение, вызванное доступом к неизвестному объекту.
yii\base\UnknownPropertyException	представляет собой исключение, вызванное доступом к свойствам неизвестного объекта.
yii\base\UserException	базовый класс для исключений, которые должны отображаться конечным пользователям.
yii\base\View	представляет объект «вид» в шаблоне MVC.
yii\base\ViewContextInterface	интерфейс, который должен реализовывать классы, которые хотят поддерживать относительные виды имен.
yii\base\ViewEvent	представляет события, инициируемые компонентом yii\base\View
yii\base\ViewNotFoundException	представляет собой исключение, вызванное отсутствием файла вида.
yii\base\ViewRenderer	базовый класс для классов рендеринга вида.
yii\base\Widget	базовый класс для виджетов.
yii\base\WidgetEvent	представляет параметр события, используемый для события виджета.
yii\behaviors\AttributeBehavior	автоматически присваивает указанное значение одному или нескольким атрибутам объекта ActiveRecord, когда происходят определенные события.
yii\behaviors\AttributeTypecastBehavior	обеспечивает возможность автоматического присваивания атрибутов модели.
yii\behaviors\AttributesBehavior	автоматически присваивает значения, указанные для одного или нескольких атрибутов объекта ActiveRecord, когда происходят

	определенные события.
yii\behaviors\BlameableBehavior	автоматически заполняет указанные атрибуты текущим идентификатором пользователя.
yii\behaviors\SluggableBehavior	автоматически заполняет указанный атрибут значением, которое можно использовать в URL-адресе.
yii\behaviors\TimestampBehavior	автоматически заполняет указанные атрибуты текущей меткой времени.
yii\caching\ApcCache	обеспечивает кэширование APC с точки зрения компонента приложения.
yii\caching\ArrayCache	обеспечивает кэширование текущего запроса только путем хранения значений в массиве.
yii\caching\Cache	базовый класс для классов кэша, поддерживающих различные реализации кэширования.
yii\caching\CacheInterface	базовый интерфейс для кэша.
yii\caching\ChainedDependency	представляет собой зависимость, которая состоит из списка других зависимостей.
yii\caching\DbCache	реализует компонент приложения кэширования, сохраняя кешированные данные в базе данных.
yii\caching\DbDependency	представляет зависимость, основанную на результате запроса оператора SQL.
yii\caching\DbQueryDependency	представляет зависимость, основанную на результате запроса экземпляра yii\db\QueryInterface
yii\caching\Dependency	базовый класс для классов зависимостей кэша.
yii\caching\DummyCache	компонент кэша заполнителя.
yii\caching\ExpressionDependency	представляет зависимость, основанную на результате выражения PHP.
yii\caching\FileCache	реализует компонент кэша с использованием файлов.
yii\caching\FileDependency	представляет зависимость, основанную на последнем времени модификации файла.
yii\caching\MemCache	реализует компонент приложения кэша на основе memcache и memcached .
yii\caching\MemCacheServer	представляет данные конфигурации для одного сервера memcache или memcached.
yii\caching\TagDependency	связывает элемент кэшированных данных с одним или несколькими \$tags .
yii\caching\WinCache	обеспечивает кэширование Windows Cache с точки зрения компонента приложения.
yii\caching\XCache	обеспечивает кеширование XCache с точки зрения компонента приложения.
yii\caching\ZendDataCache	обеспечивает кэширование данных Zend с точки зрения компонента приложения.
yii\captcha\Captcha	отображает изображение CAPTCHA и поле ввода, которое вводит введенный пользователем код подтверждения.
yii\captcha\CaptchaAction	отображает изображение CAPTCHA.
yii\captcha\CaptchaAsset	этот пакет активов предоставляет файлы javascript, необходимые для виджета yii\captcha\Captcha
yii\captcha\CaptchaValidator	подтверждает, что значение атрибута совпадает с кодом подтверждения, отображаемым в CAPTCHA.

yii\console\Application	представляет собой консольное приложение.
yii\console\Controller	базовый класс классов команд консоли.
yii\console\ErrorHandler	обрабатывает неперехваченные ошибки и исключения PHP.
yii\console\Exception	представляет собой исключение, вызванное неправильным использованием команды консоли.
yii\console\ExitCode	этот класс предоставляет константы для определения кодов выхода команд консоли.
yii\console\Markdown	анализатор Markdown, который увеличивает уценку для чтения в консольных средах.
yii\console\Request	запрос консоли представляет информацию о среде для консольного приложения.
yii\console\Response	представляет результат консольного приложения.
yii\console\UnknownCommandException	представляет собой исключение, вызванное неправильным использованием команды консоли.
yii\console\controllers\AssetController	Позволяет комбинировать и сжимать ваши файлы JavaScript и CSS.
yii\console\controllers\BaseMigrateController	базовый класс для контроллеров миграции.
yii\console\controllers\CacheController	Позволяет сбросить кеш.
yii\console\controllers\FixtureController	Управляет загрузкой и разгрузкой данных прибора.
yii\console\controllers\HelpController	Предоставляет справочную информацию о командах консоли.
yii\console\controllers\MessageController	Извлекает сообщения для перевода из исходных файлов.
yii\console\controllers\MigrateController	Управляет миграциями приложений.
yii\console\controllers\ServeController	Запускает встроенный веб-сервер PHP.
yii\console\widgets\Table	Класс Table отображает таблицу в консоли.
yii\data\ActiveDataFilter	позволяет составить условие фильтрации в формате, подходящем для yii\db\QueryInterface::where() .
yii\data\ActiveDataProvider	реализует поставщика данных на основе yii\db\Query и yii\db\ActiveQuery .
yii\data\ArrayDataProvider	реализует поставщик данных на основе массива данных.
yii\data\BaseDataProvider	предоставляет базовый класс, который реализует yii\data\DataProviderInterface .
yii\data\DataFilter	специальная yii\base\Model для обработки спецификации фильтрации запросов.
yii\data\DataProviderInterface	интерфейс, который должен быть реализован классами поставщика данных.
yii\data\Pagination	представляет информацию, относящуюся к разбивке на страницы элементов данных.
yii\data\Sort	сортировка представляет информацию, относящуюся к сортировке.
yii\data\SqlDataProvider	реализует поставщика данных на основе простого оператора SQL.
yii\db\ActiveQuery	представляет собой запрос БД, связанный ActiveRecord классом.

yii\db\ActiveQueryInterface	определяет общий интерфейс, который должен быть реализован с помощью active record query классов.
yii\db\ActiveQueryTrait	реализует общие методы и свойства для active record query классов.
yii\db\ActiveRecord	базовый класс для классов, представляющих реляционные данные в терминах объектов.
yii\db\ActiveRecordInterface	ActiveRecord интерфейс.
yii\db\ActiveRelationTrait	реализует общие методы и свойства для active record relational queries.
yii\db\AfterSaveEvent	представляет информацию, доступную в yii\db\ActiveRecord::EVENT_AFTER_INSERT и yii\db\ActiveRecord::EVENT_AFTER_UPDATE .
yii\db\BaseActiveRecord	это базовый класс для классов, представляющих реляционные данные в терминах объектов.
yii\db\BatchQueryResult	представляет пакетный запрос, из которого вы можете получать данные в пакетах.
yii\db\CheckConstraint	представляет метаданные ограничения CHECK таблицы.
yii\db\ColumnSchema	класс описывает метаданные столбца в таблице базы данных.
yii\db\ColumnSchemaBuilder	помогает определять типы схем базы данных с помощью интерфейса PHP.
yii\db\Command	представляет собой оператор SQL, который должен выполняться в базе данных.
yii\db\Connection	представляет собой соединение с базой данных через PDO .
yii\db\Constraint	представляет собой метаданные ограничения таблицы.
yii\db\ConstraintFinderTrait	предоставляет методы для получения информации ограничения таблицы.
yii\db\DataReader	представляет поток строк только для прямых из набора результатов запроса.
yii\db\DefaultValueConstraint	представляет метаданные таблицы DEFAULT
yii\db\Exception	представляет собой исключение, вызванное некоторыми операциями, связанными с БД.
yii\db\Expression	представляет собой выражение DB, которое не требует экранирования или цитирования.
yii\db\ForeignKeyConstraint	представляет метаданные таблицы ограничения FOREIGN KEY
yii\db/IndexConstraint	представляет метаданные ограничения таблицы INDEX
yii\db\IntegrityException	представляет собой исключение, вызванное нарушением ограничений DB.
yii\db\Migration	базовый класс для представления миграции базы данных.
yii\db\MigrationInterface	определяет минимальный набор методов, которые должны быть реализованы при миграции базы данных.
yii\db\Query	Запрос представляет собой инструкцию SELECT SQL таким образом, чтобы не зависеть от СУБД.
yii\db\QueryBuilder	создает инструкцию SELECT SQL на основе спецификации, заданной как объект yii\db\Query
yii\db\QueryInterface	определяет минимальный набор методов, которые будут реализованы запросом базы данных.
yii\db\QueryTrait	представляет собой минимальный набор методов запроса базы данных.

yii\db\Schema	Схема является базовым классом для конкретных классов схемы, специфичных для СУБД.
yii\db\SchemaBuilderTrait	содержит ярлыки для создания экземпляров yii\db\ColumnSchemaBuilder .
yii\db\SqlToken	представляет собой SQL-токены, созданные yii\db\SqlTokenizer или его дочерние классы.
yii\db\SqlTokenizer	разбивает SQL-запрос на отдельные токены SQL.
yii\db\StaleObjectException	
yii\db\TableSchema	представляет метаданные таблицы базы данных.
yii\db\Transaction	представляет собой транзакцию БД.
yii\db\ViewFinderTrait	реализует метод getViewNames для поиска представлений в базе данных.
yii\db\cubrid\ColumnSchemaBuilder	построитель схем для баз данных Cubrid.
yii\db\cubrid\QueryBuilder	это построитель запросов для баз данных CUBRID (версия 9.3.x и выше).
yii\db\cubrid\Schema	класс для извлечения метаданных из базы данных CUBRID (версия 9.3.x и выше).
yii\db\mssql\PDO	Это расширение стандартного PDO-класса драйверов MSSQL и DBLIB.
yii\db\mssql\QueryBuilder	построитель запросов для баз данных MS SQL Server (версия 2008 и выше).
yii\db\mssql\Schema	Схема - это класс для извлечения метаданных из баз данных MS SQL Server (версия 2008 и выше).
yii\db\mssql\SqlsrvPDO	Это расширение стандартного PDO-класса драйвера SQLSRV.
yii\db\mssql\TableSchema	представляет метаданные таблицы базы данных.
yii\db\mysql\ColumnSchemaBuilder	построитель схем для баз данных MySQL.
yii\db\mysql\QueryBuilder	построитель запросов для баз данных MySQL.
yii\db\mysql\Schema	Схема - это класс для извлечения метаданных из базы данных MySQL (версии 4.1.x и 5.x).
yii\db\oci\ColumnSchemaBuilder	построитель схем для баз данных Oracle.
yii\db\oci\QueryBuilder	построитель запросов для баз данных Oracle.
yii\db\oci\Schema	Схема - это класс для извлечения метаданных из базы данных Oracle.
yii\db\pgsql\QueryBuilder	построитель запросов для баз данных PostgreSQL.
yii\db\pgsql\Schema	Схема - это класс для извлечения метаданных из базы данных PostgreSQL (версия 9.x и выше).
yii\db\sqlite\ColumnSchemaBuilder	построитель схем для баз данных Sqlite.
yii\db\sqlite\QueryBuilder	построитель запросов для баз данных SQLite.
yii\db\sqlite\Schema	Схема - это класс для извлечения метаданных из базы данных SQLite (2/3).
yii\db\sqlite\SqlTokenizer	разбивает SQLite-запрос на отдельные токены SQL.
yii\di\Container	реализует контейнер для dependency injection

yii\di\Instance	представляет ссылку на именованный объект в контейнере инъекции зависимостей (DI) или локаторе службы.
yii\di\NotInstantiableException	представляет собой исключение, вызванное неправильной конфигурацией или использованием контейнера инъекций зависимостей.
yii\di\ServiceLocator	реализует service locator .
yii\filters\AccessControl	обеспечивает простой контроль доступа на основе набора правил.
yii\filters\AccessRule	Этот класс представляет правило доступа, определяемое фильтром действий yii\filters\AccessControl
yii\filters\AjaxFilter	позволяет ограничить доступ только для ajax-запросов.
yii\filters\ContentNegotiator	поддерживает согласование формата ответа и согласование языка приложения.
yii\filters\Cors	фильтр Cross Origin Resource Sharing .
yii\filters\HostControl	обеспечивает простой контроль над запрошенным именем хоста.
yii\filters\HttpCache	реализует кэширование на стороне клиента, используя Last-Modified и ETag HTTP заголовки.
yii\filters\PageCache	реализует серверное кэширование целых страниц.
yii\filters\RateLimitInterface	интерфейс, который может быть реализован объектом идентификации для обеспечения ограничения скорости.
yii\filters\RateLimiter	реализует алгоритм ограничения скорости, основанный на leaky bucket algorithm .
yii\filters\VerbFilter	фильтр действий, который фильтрует методы HTTP-запроса.
yii\filters\auth\AuthInterface	интерфейс, который должен быть реализован классами методов auth.
yii\filters\auth\AuthMethod	базовый класс, реализующий интерфейс yii\filters\auth\AuthInterface
yii\filters\auth\CompositeAuth	фильтр действий, который поддерживает несколько методов аутентификации одновременно.
yii\filters\auth\HttpBasicAuth	фильтр действий, поддерживающий метод базовой аутентификации HTTP.
yii\filters\auth\HttpBearerAuth	фильтр действий, который поддерживает метод аутентификации на основе токена HTTP-носителя.
yii\filters\auth\QueryParamAuth	фильтр действий, который поддерживает аутентификацию на основе токена доступа, прошедшего через параметр запроса.
yii\grid\ActionColumn	столбец для виджета yii\grid\GridView который отображает кнопки для просмотра и управления элементами.
yii\grid\CheckboxColumn	отображает столбец флажков в виде сетки.
yii\grid\Column	базовый класс всех классов столбцов yii\grid\GridView
yii\grid\DataColumn	тип столбца по умолчанию для виджета yii\grid\GridView
yii\grid\GridView	Виджет GridView используется для отображения данных в сетке.
yii\grid\GridViewAsset	Это пакет ассетов предоставляет файлы javascript для виджета yii\grid\GridView
yii\grid\RadioButtonColumn	отображает столбец переключателей в grid view.
yii\grid\SerialColumn	отображает столбец номеров строк (1-based).
yii\helpers\ArrayHelper	предоставляет дополнительные функции массива, которые вы можете использовать в своем приложении.
yii\helpers\BaseArrayHelper	обеспечивает конкретную реализацию для yii\helpers\ArrayHelper .

yii\helpers\BaseConsole	обеспечивает конкретную реализацию для yii\helpers\Console .
yii\helpers\BaseFileHelper	обеспечивает конкретную реализацию для yii\helpers\FileHelper .
yii\helpers\BaseFormatConverter	обеспечивает конкретную реализацию для yii\helpers\FormatConverter
yii\helpers\BaseHtml	обеспечивает конкретную реализацию для yii\helpers\Html .
yii\helpers\BaseHtmlPurifier	обеспечивает конкретную реализацию для yii\helpers\HtmlPurifier .
yii\helpers\BaseInflector	обеспечивает конкретную реализацию для yii\helpers\Inflector .
yii\helpers\BaseJson	обеспечивает конкретную реализацию для yii\helpers\Json .
yii\helpers\BaseMarkdown	обеспечивает конкретную реализацию для yii\helpers\Markdown .
yii\helpers\BaseStringHelper	обеспечивает конкретную реализацию для yii\helpers\StringHelper .
yii\helpers\BaseUrl	обеспечивает конкретную реализацию для yii\helpers\Url .
yii\helpers\BaseVarDumper	обеспечивает конкретную реализацию для yii\helpers\VarDumper .
yii\helpers\Console	Помощник консоли предоставляет полезные методы для задач, связанных с командной строкой, таких как получение ввода, форматирование и раскраска.
yii\helpers\FileHelper	Помощник файловой системы.
yii\helpers\FormatConverter	обеспечивает функциональность для преобразования между различными форматами шаблонов форматирования.
yii\helpers\Html	предоставляет набор статических методов для генерации часто используемых тегов HTML.
yii\helpers\HtmlPurifier	предоставляет возможность очистки HTML от любого вредоносного кода.
yii\helpers\Inflector	Инфлексор плюрализует и объединяет в себе английские существительные. Он также содержит некоторые другие полезные методы.
yii\helpers\Json	вспомогательный класс, обеспечивающий кодирование и декодирование данных JSON.
yii\helpers\Markdown	предоставляет возможность преобразования меток в HTML.
yii\helpers\ReplaceArrayValue	Объект, который представляет замену значения массива при выполнении yii\helpers\ArrayHelper::merge() .
yii\helpers\StringHelper	Помощник по строкам.
yii\helpers\UnsetArrayValue	Объект, представляющий удаление значения массива во время выполнения yii\helpers\ArrayHelper::merge() .
yii\helpers\Url	Url предоставляет набор статических методов для управления URL-адресами.
yii\helpers\VarDumper	предназначен для замены багги-функции PHP var_dump и print_r.
yii\i18n\DbMessageSource	расширяет yii\i18n\MessageSource и представляет источник сообщений, в котором хранятся переведенные сообщения в базе данных.
yii\i18n\Formatter	предоставляет набор широко используемых методов форматирования данных.
yii\i18n\GettextFile	базовый класс для представления файла сообщения Gettext.
yii\i18n\GettextMessageSource	представляет источник сообщений, основанный на GNU Gettext.
yii\i18n\GettextMoFile	представляет файл MO Gettext.
yii\i18n\GettextPoFile	представляет файл PO Gettext.
yii\i18n\I18N	I18N предоставляет функции, связанные с интернационализацией

	(I18N) и локализацией (L10N).
<code>yii\i18n\MessageFormatter</code>	позволяет форматировать сообщения через ICU message format .
<code>yii\i18n\MessageSource</code>	является базовым классом для классов репозитория сообщений.
<code>yii\i18n\MissingTranslationEvent</code>	представляет параметр для yii\i18n\MessageSource::EVENT_MISSING_TRANSLATION
<code>yii\i18n\PhpMessageSource</code>	представляет собой источник сообщений, в котором хранятся переведенные сообщения в PHP-скриптах.
<code>yii\log\DbTarget</code>	хранит сообщения журнала в таблице базы данных.
<code>yii\log\Dispatcher</code>	диспетчер управляет набором log targets .
<code>yii\log>EmailTarget</code>	отправляет выбранные сообщения журнала на указанные адреса электронной почты.
<code>yii\log\FileTarget</code>	записывает сообщения журнала в файл.
<code>yii\log\Logger</code>	регистратор регистрирует зарегистрированные сообщения в памяти и отправляет их на разные цели, если \$dispatcher установлен.
<code>yii\log\SyslogTarget</code>	записывает журнал в syslog.
<code>yii\log\Target</code>	базовый класс для всех целевых классов журнала.
<code>yii\mail\BaseMailer</code>	служит базовым классом, который реализует основные функции, требуемые yii\mail\MailerInterface .
<code>yii\mail\BaseMessage</code>	служит базовым классом, который реализует метод send() требуемый yii\mail\MessageInterface .
<code>yii\mail\MailEvent</code>	представляет параметр события, используемый для событий, вызванных yii\mail\BaseMailer .
<code>yii\mail\MailerInterface</code>	интерфейс, который должен реализовываться классами почтовой программы.
<code>yii\mail\MessageInterface</code>	интерфейс, который должен быть реализован классами почтовых сообщений.
<code>yii\mutex\DbMutex</code>	базовый класс для классов, который использует базу данных при реализации механизма блокировки mutex.
<code>yii\mutex\FileMutex</code>	реализует механизм блокировки mutex через файлы локальной файловой системы.
<code>yii\mutex\Mutex</code>	Компонент Mutex позволяет взаимное выполнение параллельных процессов, чтобы предотвратить "race conditions".
<code>yii\mutex\MysqlMutex</code>	реализовал механизм блокировки mutex через блокировки MySQL.
<code>yii\mutex\OracleMutex</code>	реализовал механизм блокировки mutex через блокировки Oracle.
<code>yii\mutex\PgsqlMutex</code>	реализовал механизм блокировки mutex через блокировки PostgreSQL.
<code>yii\rbac\Assignment</code>	Назначение представляет собой назначение роли пользователю.
<code>yii\rbac\BaseManager</code>	базовый класс, реализующий yii\rbac\ManagerInterface для управления RBAC.
<code>yii\rbac\CheckAccessInterface</code>	Подробно см. в разделе guide article on security authorization .
<code>yii\rbac\DbManager</code>	представляет собой диспетчер авторизации, который хранит информацию авторизации в базе данных.
<code>yii\rbac\Item</code>	Подробно см. в разделе guide article on security authorization .
<code>yii\rbac\ManagerInterface</code>	Подробно см. в разделе guide article on security authorization .
<code>yii\rbac\Permission</code>	Подробно см. в разделе guide article on security authorization .
<code>yii\rbac\PhpManager</code>	представляет собой диспетчер авторизации, который хранит информацию авторизации в терминах файла сценария PHP.

yii\rbac\Role	Подробно см. в разделе guide article on security authorization .
yii\rbac\Rule	Правило представляет собой бизнес-ограничение, которое может быть связано с ролью, разрешением или назначением.
yii\rest\Action	базовый класс для классов действий, реализующих API RESTful.
yii\rest\ActiveController	реализует общий набор действий для поддержки доступа RESTful к ActiveRecord.
yii\rest\Controller	является базовым классом для классов контроллера API RESTful.
yii\rest\CreateAction	реализует конечную точку API для создания новой модели из данных.
yii\rest>DeleteAction	реализует конечную точку API для удаления модели.
yii\rest\IndexAction	реализует конечную точку API для перечисления нескольких моделей.
yii\rest\OptionsAction	отвечает на запрос OPTIONS, отправив обратно заголовок <code>Allow</code>
yii\rest\Serializer	преобразует объекты ресурсов и коллекции в представление массива.
yii\rest\UpdateAction	реализует конечную точку API для обновления модели.
yii\rest\UrlRule	предоставляется для упрощения создания правил URL для поддержки API RESTful.
yii\rest\ViewAction	реализует конечную точку API для возврата подробной информации о модели.
yii\test\ActiveFixture	представляет собой устройство, поддерживаемое ActiveRecord class или database table .
yii\test\ArrayFixture	представляет собой произвольное устройство, которое может быть загружено из файлов PHP.
yii\test\BaseActiveFixture	базовый класс для классов приборов, которые поддерживают доступ к данным прибора как объекты ActiveRecord.
yii\test\DbFixture	базовый класс для устройств, связанных с БД.
yii\test\Fixture	представляет собой фиксированное состояние тестовой среды.
yii\test\FixtureTrait	предоставляет функции для загрузки, выгрузки и доступа к светильникам для тестового примера.
yii\test\InitDbFixture	представляет начальное состояние, необходимое для тестов, связанных с БД.
yii\validators\BooleanValidator	проверяет, является ли значение атрибута логическим значением.
yii\validators\CompareValidator	сравнивает указанное значение атрибута с другим значением.
yii\validators\DateValidator	проверяет, соответствует ли атрибут дате, времени или дате в соответствующем <code>\$format</code> .
yii\validators\DefaultValueValidator	устанавливает атрибут как заданное значение по умолчанию.
yii\validators\EachValidator	проверяет массив, проверяя каждый его элемент на встроенное правило проверки.
yii\validators\EmailValidator	подтверждает, что значение атрибута является допустимым адресом электронной почты.
yii\validators\ExistValidator	проверяет, что значение атрибута существует в таблице.
yii\validators\FileValidator	проверяет, получает ли атрибут действительный загруженный файл.
yii\validators\FilterValidator	преобразует значение атрибута в соответствии с фильтром.
yii\validators\ImageValidator	проверяет, получает ли атрибут действительное изображение.

yii\validators\InlineValidator	представляет собой валидатор, который определяется как метод в проверяемом объекте.
yii\validators\IpValidator	Валидатор проверяет, является ли значение атрибута допустимым адресом IPv4 / IPv6 или подсетью.
yii\validators\NumberValidator	подтверждает, что значение атрибута является числом.
yii\validators\PunycodAsset	Этот пакет активов предоставляет файлы javascript, необходимые для проверки клиента yii\validators\EmailValidators
yii\validators\RangeValidator	подтверждает, что значение атрибута относится к числу значений.
yii\validators\RegularExpressionValidator	подтверждает, что значение атрибута соответствует указанном \$pattern .
yii\validators\RequiredValidator	проверяет, что указанный атрибут не имеет нулевого или пустого значения.
yii\validators\SafeValidator	служит фиктивным валидатором, основная цель которого - отметить атрибуты, которые должны быть безопасными для массового присвоения.
yii\validators\StringValidator	проверяет, что значение атрибута имеет определенную длину.
yii\validators\UniqueValidator	подтверждает, что значение атрибута уникально в указанной таблице базы данных.
yii\validators\UrlValidator	подтверждает, что значение атрибута является допустимым URL-адресом http или https.
yii\validators\ValidationAsset	Этот пакет активов предоставляет файлы javascript для проверки клиента.
yii\validators\Validator	базовый класс для всех валидаторов.
yii\web\Application	является базовым классом для всех классов веб-приложений.
yii\web\AssetBundle	представляет собой набор файлов активов, таких как CSS, JS, изображения.
yii\web\AssetConverter	поддерживает преобразование нескольких популярных форматов скриптов в сценарии JS или CSS.
yii\web\AssetConverterInterface	должен быть реализован классами преобразователей.
yii\web\AssetManager	управляет конфигурацией и загрузкой пакета активов.
yii\web\BadRequestHttpException	представляет исключение HTTP «Bad Request» с кодом состояния 400.
yii\web\CacheSession	реализует компонент сеанса с использованием кеша в качестве носителя данных.
yii\web\CompositeUrlRule	базовый класс для классов правил URL, которые состоят из нескольких простых правил.
yii\web\ConflictHttpException	представляет собой исключение HTTP «Конфликт» с кодом состояния 409
yii\web\Controller	является базовым классом веб-контроллеров.
yii\web\Cookie	представляет информацию, связанную с файлом cookie, например \$name , \$value , \$domain , и т.д.
yii\web\CookieCollection	поддерживает файлы cookie, доступные в текущем запросе.
yii\web\DbSession	расширяет yii\web\Session используя базу данных в качестве хранилища данных сеанса.
yii\web>ErrorAction	отображает ошибки приложения с использованием указанного вида.
yii\web\ErrorHandler	обрабатывает перехваченные ошибки и исключения PHP.
yii\web\ForbiddenHttpException	представляет собой «Запрещенное» исключение HTTP с кодом

	состояния 403.
yii\web\GoneHttpException	представляет исключение HTTP «Gone» с кодом состояния 410.
yii\web\GroupUrlRule	представляет собой набор правил URL, имеющих один и тот же префикс в своих шаблонах и маршрутах.
yii\web\HeaderCollection	используется yii\web\Response для поддержки зарегистрированных HTTP-заголовков.
yii\web\HtmlResponseFormatter	форматирует данные в виде HTML-ответа.
yii\web\HttpException	представляет собой исключение, вызванное неправильным запросом конечного пользователя.
yii\web\IdentityInterface	интерфейс, который должен быть реализован классом, предоставляющим идентификационную информацию.
yii\web\JqueryAsset	Этот пакет ассетов предоставляет jQuery JavaScript-библиотеку.
yii\web\JsExpression	отмечает строку как выражение JavaScript.
yii\web\JsonParser	Просматривает необработанный HTTP-запрос, используя yii\helpers\Json::decode() .
yii\web\JsonResponseFormatter	форматирует данные в виде ответа JSON или JSONP.
yii\web\Link	представляет объект ссылки, определенный в JSON Hypermedia API Language .
yii\web\Linkable	интерфейс, который должен реализовываться классами, которые обычно представляют локализованные ресурсы.
yii\web\MethodNotAllowedHttpException	представляет исключение HTTP с недопустимым хостом с кодом состояния 405.
yii\web\MultiFieldSession	является базовым классом для реализации хранилищ сеансов с поддержкой многопоточного хранения данных.
yii\web\MultipartFormDataParser	анализирует содержимое, закодированное как 'multipart/form-data'.
yii\web\NotAcceptableHttpException	представляет собой «недопустимое» исключение HTTP с кодом состояния 406.
yii\web\NotFoundHttpException	представляет исключение HTTP «Not Found» с кодом состояния 404.
yii\web\RangeNotSatisfiableHttpException	представляет собой исключение, вызванное неправильным запросом конечного пользователя.
yii\web\Request	Класс Web Request представляет HTTP-запрос.
yii\web\RequestParserInterface	Интерфейс для классов, которые анализируют тело необработанного запроса в массив параметров.
yii\web\Response	Класс Web Response представляет собой HTTP-ответ.
yii\web\ResponseFormatterInterface	указывает интерфейс, необходимый для форматирования ответа до его отправки.
yii\web\ServerErrorHttpException	представляет ошибку HTTP "Internal Server Error" с кодом состояния 500.
yii\web\Session	Сеанс предоставляет управление данными сеанса и связанные с ним конфигурации.
yii\web\SessionIterator	реализует iterator для перемещения переменных сеанса, управляемых yii\web\Session .
yii\web\TooManyRequestsHttpException	представляет исключение HTTP "Too Many Requests" с кодом состояния 429.
yii\web\UnauthorizedHttpException	представляет собой «несанкционированное» исключение HTTP с кодом состояния 401.

yii\web\UnprocessableEntityHttpException	представляет исключение HTTP «Unprocessable Entity» с кодом состояния 422.
yii\web\UnsupportedMediaTypeHttpException	представляет исключение HTTP «Unsupported Media Type» с кодом статуса 415.
yii\web\UploadedFile	представляет информацию для загруженного файла.
yii\web\UrlManager	обрабатывает анализ HTTP-запросов и создает URL-адреса на основе набора правил.
yii\web\UrlNormalizer	нормализует URL-адреса для yii\web\UrlManager и yii\web\UrlRule .
yii\web\UrlNormalizerRedirectException	представляет собой информацию для перенаправления, которая должна выполняться во время нормализации URL.
yii\web\UrlRule	представляет правило, используемое yii\web\UrlManager для анализа и генерации URL-адресов.
yii\web\UrlRuleInterface	интерфейс, который должен реализовываться классами правил URL.
yii\web\User	это класс для компонента user приложения, который управляет статусом аутентификации пользователя.
yii\web\UserEvent	Этот класс событий используется для событий, инициированных классом yii\web\User
yii\web\View	Просмотр представляет объект представления в шаблоне MVC.
yii\web\ViewAction	представляет действие, которое отображает представление в соответствии с заданным пользователем параметром.
yii\web\XmlResponseFormatter	форматирует данные в XML-ответе.
yii\web\YiiAsset	Этот пакет ассетов предоставляет базовые файлы JavaScript для Yii Framework.
yii\widgets\ActiveField	представляет поле ввода формы внутри yii\widgets\ActiveForm .
yii\widgets\ActiveForm	это виджет, который создает интерактивную форму HTML для одной или нескольких моделей данных.
yii\widgets\ActiveFormAsset	Пакет активов для виджета yii\widgets\ActiveForm
yii\widgets\BaseListView	базовый класс для виджетов, отображающих данные от поставщика данных, таких как ListView и GridView.
yii\widgets\Block	Блок записывает все выходные данные между вызовами begin() и end() и сохраняет их в блоках yii\base\View::\$blocks .
yii\widgets\Breadcrumbs	отображает список ссылок, обозначающих позицию текущей страницы во всей иерархии сайтов.
yii\widgets\ContentDecorator	записывает весь вывод между вызовами begin() и end() передает его в данный файл вида в виде \$content а затем эхо-результат рендеринга.
yii\widgets\DetailView	отображает детали одной модели \$model .
yii\widgets\FragmentCache	используется yii\base\View для обеспечения кэширования фрагментов страницы.
yii\widgets\InputWidget	базовый класс для виджетов, которые собирают пользовательские входы.
yii\widgets\LinkPager	отображает список гиперссылок, которые приводят к разным страницам цели.
yii\widgets\LinkSorter	отображает список сортировочных ссылок для данного определения сортировки.
yii\widgets\ListView	Виджет используется для отображения данных из поставщика данных. Каждая модель данных создается с использованием

	указанного вида.
yii\widgets\MaskedInput	генерирует скрытый ввод текста.
yii\widgets\MaskedInputAsset	Пакет ассетов для виджета yii\widgets\MaskedInput
yii\widgets\Menu	Меню отображает многоуровневое меню с использованием вложенных списков HTML.
yii\widgets\Pjax	виджет, интегрирующий плагин pjax jQuery.
yii\widgets\PjaxAsset	Этот пакет активов предоставляет файлы javascript, необходимые для виджета yii\widgets\Pjax
yii\widgets\Spaceless	Spaceless виджит удаляет пробельные символы между тегами HTML. Пробелы в тегах HTML или в обычном тексте всегда остаются нетронутыми.

Структура приложения Yii

basic/	корневой каталог приложения
composer.json	используется Composer'ом, содержит описание приложения
config/	конфигурационные файлы
console.php	конфигурация консольного приложения
web.php	конфигурация Web приложения
commands/	содержит классы консольных команд
controllers/	контроллеры
models/	модели
runtime/	файлы, которые генерирует Yii во время выполнения приложения (логи, кэш)
vendor/	содержит пакеты Composer'а и, собственно, сам фреймворк Yii
views/	виды приложения
web/	корневая директория Web приложения. Содержит файлы, доступные через Web
assets/	скрипты, используемые приложением (js, css)
index.php	точка входа в приложение Yii. С него начинается выполнение приложения
yii	скрипт выполнения консольного приложения Yii

Говорим «Привет»

Действия объявляются в [контроллерах](#).

```
namespace app\controllers;
use yii\web\Controller;
class SiteController extends Controller
{
    // ...существующий код...
    public function actionSay($message = 'Привет')
    {
        return $this->render('say', ['message' => $message]);
    }
}
```

[Представления](#) являются скриптами, которые используются для формирования тела ответа.

```
use yii\helpers\Html;
echo Html::encode($message);
```

Работа с формами

Создание модели. В файле `models/EntryForm.php` создайте класс модели `EntryForm`

```
namespace app\models;
use yii\base\Model;
class EntryForm extends Model
{
    public $name;
    public $email;
    public function rules()
    {
        return [
            [['name', 'email'], 'required'],
            ['email', 'email'],
        ];
    }
}
```

Создание действия. Далее создайте действие `entry` в контроллере `site`.

```
namespace app\controllers;
use Yii;
use yii\web\Controller;
use app\models\EntryForm;

class SiteController extends Controller
{
    public function actionEntry()
    {
        $model = new EntryForm();
        if ($model->load(Yii::$app->request->post()) && $model->validate()) {
            return $this->render('entry-confirm', ['model' => $model]);
        } else {
            return $this->render('entry', ['model' => $model]);
        }
    }
}
```

В заключение, создаём два представления с именами `entry-confirm` и `entry`

```
<?php
use yii\helpers\Html;
?>
<p>Вы ввели следующую информацию:</p>
<ul>
    <li><label>Name</label>: <?= Html::encode($model->name) ?></li>
    <li><label>Email</label>: <?= Html::encode($model->email) ?></li>
</ul>
```

Представление entry отображает HTML форму

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;
?>
<?php $form = ActiveForm::begin(); ?>
    <?= $form->field($model, 'name') ?>
    <?= $form->field($model, 'email') ?>
    <div class="form-group">
        <?= Html::submitButton('Отправить', ['class' => 'btn btn-primary']) ?>
    </div>
<?php ActiveForm::end(); ?>
```

Чтобы увидеть всё созданное в работе, откройте в браузере следующий URL:

<http://localhost/basic/web/index.php?r=site%2Fentry>

Работа с базами данных

Откройте файл config/db.php и измените параметры на верные для вашей базы данных.

```
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=yii2basic',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
];
```

Создайте класс - потомок [Active Record](#), под названием Country, и сохраните его в файле models/Country.php

```
namespace app\models;
use yii\db\ActiveRecord;

class Country extends ActiveRecord
{
}
```

создайте действие в котором опишите вызов данных из таблицы базы данных

```
namespace app\controllers;

use yii\web\Controller;
use yii\data\Pagination;
use app\models\Country;

class SiteController extends Controller
{
    public function actionTest()
    {
        $query = Country::find();
```

```

        $pagination = new Pagination([
            'defaultPageSize' => 5,
            'totalCount' => $query->count(),
        ]);

        $countries = $query->orderBy('name')
            ->offset($pagination->offset)
            ->limit($pagination->limit)
            ->all();

        return $this->render('index', [
            'countries' => $countries,
            'pagination' => $pagination,
        ]);
    }
}

```

создайте вид который отобразит данные в пагинаторе

```

<?php
use yii\helpers\Html;
use yii\widgets\LinkPager;
?>
<h1>Countries</h1>
<ul>
<?php foreach ($countries as $country): ?>
    <li>
        <?= Html::encode("{ $country->name} ({ $country->code})") ?>:
        <?= $country->population ?>
    </li>
<?php endforeach; ?>
</ul>

<?= LinkPager::widget(['pagination' => $pagination]) ?>

```

Генерация кода при помощи Gii

Gii представлен в Yii как [модуль](#).

Если вы посмотрите [входной скрипт](#) web/index.php вашего приложения, вы увидите следующую строку, устанавливающую константу YII_ENV_DEV в значение true.

```
defined('YII_ENV') or define('YII_ENV', 'dev');
```

Благодаря этой строке ваше приложение находится в режиме разработки, и Gii уже активирован в соответствии с описанной выше конфигурацией. Теперь вы можете получить доступ к Gii по следующему адресу:

```
http://hostname/index.php?r=gii
```

Входные скрипты

Объявление констант

`YII_DEBUG`: указывает работает ли приложение в отладочном режиме.

По умолчанию значение `YII_DEBUG` равно `false`

`YII_ENV`: указывает в каком окружении запущено приложение.

По умолчанию значение `YII_ENV` равно `'prod'`, приложение запущено в производственном режиме;

`YII_ENABLE_ERROR_HANDLER`: указывает нужно ли включать обработчик ошибок.

По умолчанию значение равно `true`.

При определении константы, обычно используем следующий код

```
defined('YII_DEBUG') or define('YII_DEBUG', true);
```

Приложение

Созданное приложение доступно глобально `Yii::$app`

Обязательные свойства приложения:

`[[yii\base\Application::id|id]]` - это уникальный индекс приложения

`[[yii\base\Application::basePath|basePath]]` - указывает на корневую директорию приложения

Важные свойства:

`[[yii\base\Application::aliases|aliases]]` - позволяет настроить вам множество псевдонимов в рамках массива

```
[
    'aliases' => [
        '@name1' => 'path/to/path1',
        '@name2' => 'path/to/path2',
    ],
]
```

`[[yii\base\Application::bootstrap|bootstrap]]` - позволяет указать массив компонентов, которые должны быть загружены при старте приложения

`[[yii\web\Application::catchAll|catchAll]]` - указывает действие контроллера, которое должно обрабатывать все входящие запросы от пользователя.

```
[
    'catchAll' => [
        'offline/notice',
        'param1' => 'value1',
        'param2' => 'value2',
    ],
]
```


`[[yii\base\Application::components|components]]` - ПОЗВОЛЯЕТ

зарегистрировать список именованных компонентов. `Yii::$app->componentID`

```
[
    'components' => [
        'cache' => [
            'class' => 'yii\caching\FileCache',
        ],
        'user' => [
            'identityClass' => 'app\models\User',
            'enableAutoLogin' => true,
        ],
    ],
]
```

`[[yii\base\Application::controllerMap|controllerMap]]` - ПОЗВОЛЯЕТ ВАМ

задавать соответствия между ID контроллера и произвольным классом контроллера.

```
[
    'controllerMap' => [
        'account' => 'app\controllers\UserController',
        'article' => [
            'class' => 'app\controllers\PostController',
            'enableCsrfValidation' => false,
        ],
    ],
]
```

`[[yii\base\Application::controllerNamespace|controllerNamespace]]` -

указывает пространство имен, в котором по умолчанию должны находиться названия классов контроллеров. (по умолчанию `app\controllers`)

`[[yii\base\Application::language|language]]` - указывает язык

приложения, на котором содержимое страницы должно быть отображено (для английского в США: `en-US`, для русского в России `ru-RU`)

`[[yii\base\Application::modules|modules]]` - указывает модули, которые содержатся в приложении.

```
[
    'modules' => [
        'booking' => 'app\modules\booking\BookingModule',
        'comment' => [
            'class' => 'app\modules\comment\CommentModule',
            'db' => 'db',
        ],
    ],
]
```

`[[yii\base\Application::name|name]]` - указывает название приложения, которое может быть показано конечным пользователям

`[[yii\base\Application::params|params]]` - указывает массив глобально доступных параметров приложения.

```
[
    'params' => [ 'thumbnail.size' => [128, 128], ],
]
```

Пример использования параметров

```
$size = \Yii::$app->params['thumbnail.size'];  
$width = \Yii::$app->params['thumbnail.size'][0];
```

`[[yii\base\Application::sourceLanguage|sourceLanguage]]` — указывает язык на котором написан код приложения. По-умолчанию 'en-US'

`[[yii\base\Application::timeZone|timeZone]]` — предоставляет альтернативный способ установки временной зоны

```
[  
    // Europe/Moscow для России (прим. пер.)  
    'timeZone' => 'America/Los_Angeles',  
]
```

`[[yii\base\Application::version|version]]` - указывает версию приложения. (По умолчанию 1.0)

Полезные свойства:

`[[yii\base\Application::charset|charset]]` - указывает кодировку, которую использует приложение. (UTF-8)

`[[yii\base\Application::defaultRoute|defaultRoute]]` - казывает маршрут, который должно использовать приложение, когда он не указан во входящем запросе.

`[[yii\base\Application::extensions|extensions]]` - список расширений, которые установлены и используются приложением.

`[[yii\base\Application::layout|layout]]` - указывает имя шаблона по умолчанию, который должен быть использовать при формировании представлений.

`[[yii\base\Application::layoutPath|layoutPath]]` - указывает путь, по которому следует искать шаблоны.

`[[yii\base\Application::runtimePath|runtimePath]]` - указывает путь, по которому хранятся временные файлы

`[[yii\base\Application::viewPath|viewPath]]` - указывает базовую папку, где содержаться все файлы представлений.

`[[yii\base\Application::vendorPath|vendorPath]]` - казывает папку сторонних библиотек, которые используются и управляются Composer.

`[[yii\console\Application::enableCoreCommands|enableCoreCommands]]` - указывает нужно ли использовать встроенные в Yii консольные команды.

События приложения:

Назначать обработчики событий в конфигурации приложения следующим образом:

```
[  
    'on beforeRequest' => function ($event) {  
        // ...  
    },  
]
```

Также вы можете назначить обработчики событий в процессе начальной загрузки приложения, сразу после того как приложение будет создано.

```
\Yii::$app->on(\yii\base\Application::EVENT_BEFORE_REQUEST, function ($event) {
    // ...
});
[[yii\base\Application::EVENT_BEFORE_REQUEST|EVENT_BEFORE_REQUEST]] -
    событие возникает до того как приложение начинает обрабатывать
    входящий запрос. beforeRequest

[[yii\base\Application::EVENT_AFTER_REQUEST|EVENT_AFTER_REQUEST]] -
    событие возникает после того как приложение заканчивает обработку
    запроса, но до того как произойдет отправка ответа. afterRequest

[[yii\base\Application::EVENT_BEFORE_ACTION|EVENT_BEFORE_ACTION]] -
    событие возникает до того как будет выполнено действие контроллера.
    beforeAction

[
    'on beforeAction' => function ($event) {
        if (некоторое условие) {
            $event->isValid = false;
        } else {
        }
    },
]

[[yii\base\Application::EVENT_AFTER_ACTION|EVENT_AFTER_ACTION]] -
    событие возникает после выполнения действия контроллера. afterAction

[
    'on afterAction' => function ($event) {
        if (некоторое условие) {
            // modify $event->result
        } else {
        }
    },
]
```

Компоненты приложения

Созданное приложение доступно глобально `Yii::$app->componentID`

Компонентами приложения могут быть любые объекты. Вы можете зарегистрировать их с помощью свойства `[[yii\base\Application::components]]` в конфигурации приложения.

```
[
    'components' => [
        // регистрация "cache" компонента с помощью имени класса
        'cache' => 'yii\caching\ApcCache',

        // регистрация "search" компонента с помощью анонимной функции
        'search' => function () {
            return new app\components\SolrService;
        },
    ],
]
```

Компоненты начальной загрузки:

Необходимо указать ID компонента в качестве элемента свойства

```
[ [yii\base\Application::bootstrap|bootstrap] ]
```

К примеру, при данной конфигурации компонент log всегда подгружается при загрузке:

```
[
    'bootstrap' => [
        'log',
    ],
    'components' => [
        'log' => [
            // конфигурация для компонента `log`
        ],
    ],
]
```

Встроенные компоненты приложения:

[[yii\web\AssetManager|assetManager]]: используется для управления и опубликования ресурсов приложения.

[[yii\db\Connection|db]]: представляет собой соединение с базой данных

[[yii\base\Application::errorHandler|errorHandler]]: осуществляет обработку PHP ошибок и исключений.

[[yii\i18n\Formatter|formatter]]: форматирует данные для отображения их конечному пользователю.

[[yii\i18n\I18N|i18n]]: используется для перевода сообщений и форматирования.

[[yii\log\Dispatcher|log]]: обработка и маршрутизация логов.

[[yii\swiftmailer\Mailer|mailer]]: предоставляет возможности для составления и рассылки писем.

[[yii\base\Application::response|response]]: представляет собой данные от сервера, которые будут направлены пользователю.

[[yii\base\Application::request|request]]: представляет собой запрос, полученный от конечных пользователей.

[[yii\web\Session|session]]: информация о сессии.

[[yii\web\UrlManager|urlManager]]: используется для разбора и создания URL.

[[yii\web\User|user]]: представляет собой информацию аутентифицированного пользователя.

[[yii\web\View|view]]: используется для отображения представлений.

Контроллеры

Контроллеры являются частью MVC архитектуры. (Котроллер → Модель → Вид)

Объекты классов, унаследованных от `[[yii\base\Controller]]`

Контроллеры состоят из действий.

Создание действий:

```
namespace app\controllers;

use yii\web\Controller;

class SiteController extends Controller
{
    public function actionIndex()
    {
        return $this->render('index');
    }

    public function actionHelloWorld()
    {
        return 'Hello World';
    }
}
```

Правила наименования классов контроллеров

article соответствует `app\controllers\ArticleController`;

post-comment соответствует `app\controllers\PostCommentController`;

admin/post-comment соответствует `app\controllers\admin\PostCommentController`;

Карта контроллеров

Вы можете сконфигурировать в настройках приложения следующим образом:

```
[
    'controllerMap' => [
        // объявляет "account" контроллер, используя название класса
        'account' => 'app\controllers\UserController',

        // объявляет "article" контроллер, используя массив конфигурации
        'article' => [
            'class' => 'app\controllers\PostController',
            'enableCsrfValidation' => false,
        ],
    ],
]
```

Контроллер по умолчанию

Каждое приложение имеет контроллер по умолчанию, в настройках приложения

```
[
    'defaultRoute' => 'main',
]
```


Отдельные действия

Определяются в качестве классов, унаследованных от `[[yii\base\Action]]`

```
namespace app\components;
use yii\base\Action;

class HelloWorldAction extends Action
{
    public function run()
    {
        return "Hello World";
    }
}
```

Для использования отдельного действия, вы должны указать его в карте действий

```
public function actions()
{
    return [
        // объявляет "error" действие с помощью названия класса
        'error' => 'yii\web\ErrorAction',

        // объявляет "view" действие с помощью конфигурационного массива
        'view' => [
            'class' => 'yii\web\ViewAction',
            'viewPrefix' => '',
        ],
    ];
}
```

Результаты действий

