

# Программно-аппаратные средства Web

## CMF Drupal

Сергей Геннадьевич Синица

КубГУ, 2020

[sin@kubsu.ru](mailto:sin@kubsu.ru)

# Drupal™

We're the leading open-source CMS for ambitious digital experiences that reach your audience across multiple channels.

Because we all have different needs, Drupal allows you to create a unique space in a world of cookie-cutter solutions.



## Drupal for **Developers**

*Build the open web  
with Drupal*



## Drupal for **Marketers**

*In a world full of templates,  
be original*

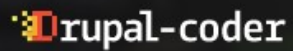


## Drupal for **Agencies**

*Achieve your  
clients' ambition*

<https://drupal-coder.ru/>

Инитлаб в России + Claios GmbH в Германии



[DRUPAL SUPPORT](#)

[TEAM](#)

[DRUPALGIVE](#)

[SERVICES ▾](#)

[CUSTOMERS](#)

[CONTACTS](#)

# Drupal development

Development and support of  
Drupal-powered websites since 2005

[PLANS](#)

**#1** 

Drupal developer  
in the Runet Rating

**3+**

our specialists have 3+  
years of Drupal  
experience  
on average

**14**

years of Drupal-related  
experience

**50+**

DrupalGive themes  
and modules

**90 000+**

hours supporting  
Drupal-powered sites

**300+**

sites on  
support



# одно из крупнейших в мире OpenSource сообществ

100k+ разработчиков  
30k+ модулей  
2000+ шаблонов  
1000+ сборок

## Drupal.org Activity

**54,037** people with Git accounts  
**2,431** Git commits this week  
**1,406,025** users on drupal.org  
**1,149,258** sites running Drupal  
**8,280,071** comments and issue followups









**DrupalCon**  
GLOBAL 2020

DIAMOND SPONSORS



Log in

Next event  
**GLOBAL 2020**  
JULY 14-17, 2020

[Program](#) [Register](#) [Sponsor](#) [About](#)

[Register Now](#)

# DrupalCon

GLOBAL 2020:  
THE OPEN SOURCE DIGITAL EXPERIENCE CONFERENCE  
JULY 14-17 | ONLINE EVERYWHERE

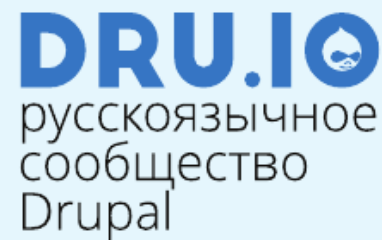
[REGISTER NOW](#)

----- BROUGHT TO YOU BY -----



# Drupal в России

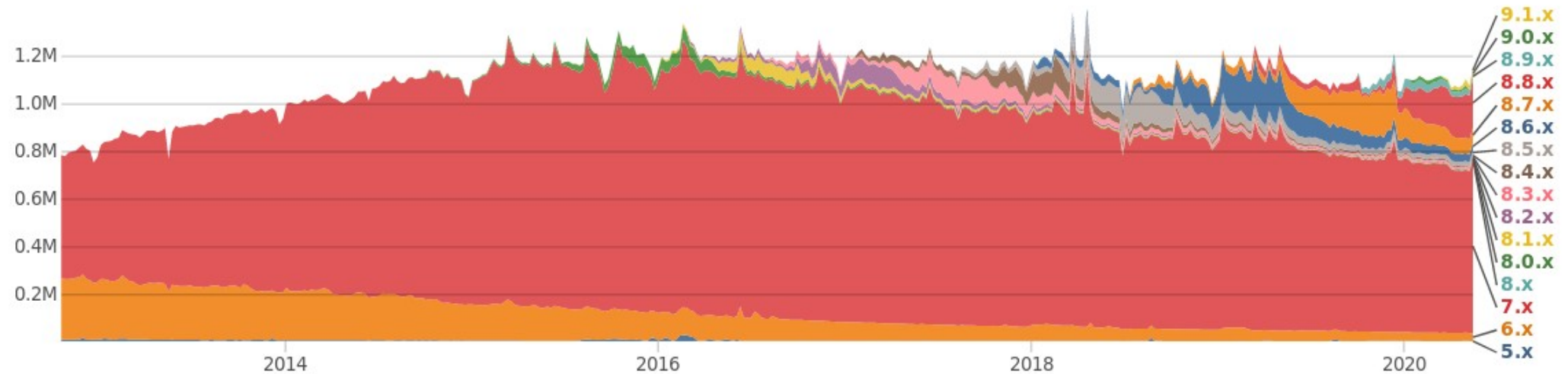
- Более 500 профессиональных разработчиков
- Конференции в Москве, Санкт-Петербурге, Новосибирске, Краснодаре
- Более 50 локальных сообществ в разных городах



[t.me/drupal\\_rus](https://t.me/drupal_rus) и [t.me/drupal\\_beginner](https://t.me/drupal_beginner) чаты Telegram

# Версии Drupal и доля рынка

Weekly project usage



TOP3 CMS в мире

Drupal 9 релиз 3 июня 2020!


TOP10 в Ecommerce (Ubercart)

TOP1 в Fortune 500 или Alexa top 10k

Министерство Юстиции Российской Федерации, король Бельгии, Белый Дом США и NASA  
Tesla, правительство Австралии для всех государственных сайтов страны, Сбербанк,  
Всемирный Банк, UNESCO, Unicef, PayPal, FedEx, ebay, Джонсон и Джонсон, Nestle.

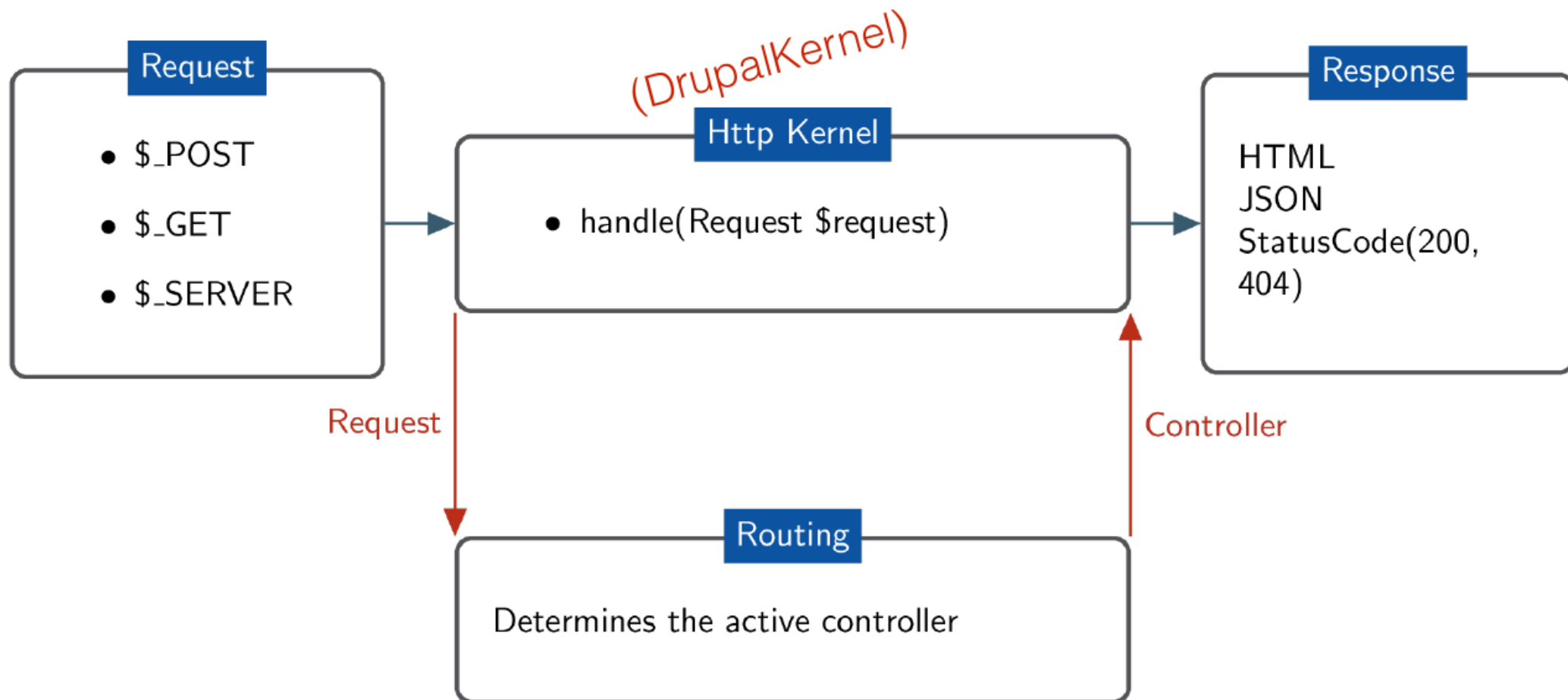


# Drupal 8

1.  Symfony components
2. 
3. 

# Symfony components

- [ClassLoader](#)
- [Console](#)
- [CssSelector](#)
- [DependencyInjection](#)
- [EventDispatcher](#)
- [HttpFoundation](#)
- [HttpKernel](#)
- [Process](#)
- [Routing](#)
- [Serializer](#)
- [Translation](#)
- [Validator](#)
- [Yaml](#)








modules/example/src/Controller/ExampleController.php

```
class ExampleController extends ControllerBase {  
  
    /**  
     * {@inheritdoc}  
     */  
    public function content() {  
        $build = array(  
            '#type' => 'markup',  
            '#markup' => t('Hello World!'),  
        );  
        return $build;  
    }  
  
}
```

modules/example/example.routing.yml

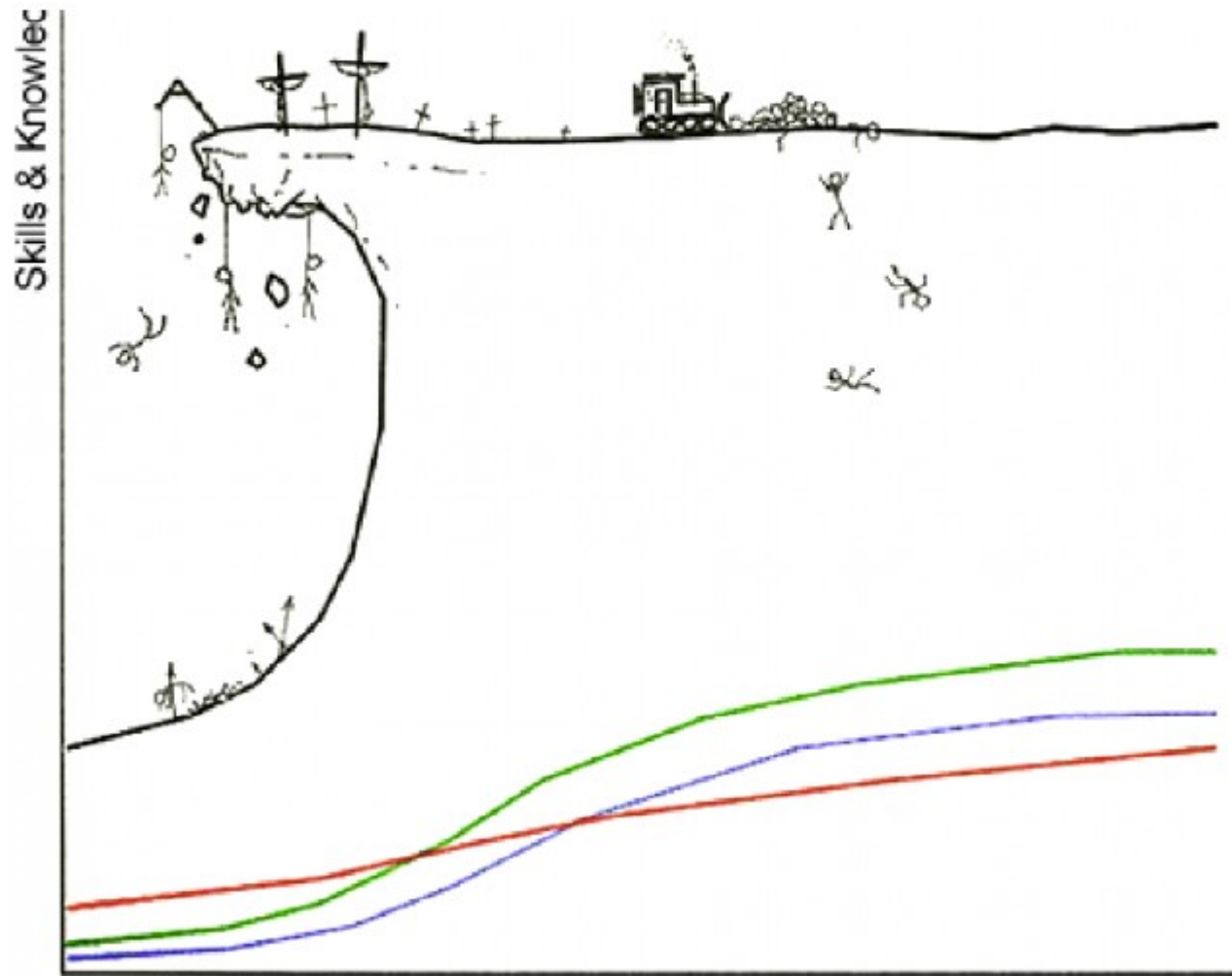
```
example.content:  
  path: '/example'  
  defaults:  
    _controller: '\Drupal\example\Controller\ExampleController::content'  
    _title: 'Hello World'  
  requirements:  
    _permission: 'access content'
```

# Drupal 8

1.  Symphony components
2.  Drupal traditions
3. 



# Drupal Traditions



[illegible]

<https://www.drupal.org/developing/api/8/render/pipeline>

<https://drupal-coder.ru/drupal-courses>

РАЗРАБОТКА **МОДУЛЕЙ**

**Drupal™ 8**

Подробнее

РАЗРАБОТКА **МОДУЛЕЙ**

**Drupal™ 7**

РАЗРАБОТКА **САЙТОВ**

**Drupal™ 8**

РАЗРАБОТКА **САЙТОВ**

**Drupal™ 7**



# Что писать в модулях

1. Routing
2. Services
3. Events
4. Entities
5. Plugins
6. Hooks

# СУЩНОСТИ

## **Content Entities**

- Customized fields
- Stored in database tables (by default)
- Mostly created on front-end

### **Examples**

- Nodes
- Custom Blocks
- Users
- Comments
- Taxonomy Terms
- Menu Links
- Aggregator Feeds/Items

## **Config Entities**

- Deployed to different environments
- Stored in configuration system
- Mostly created on back-end

### **Examples**

- Content Types
- Custom Block Types
- User Roles
- Views
- Taxonomy Vocabularies
- Menus
- Image Styles




# Всё что накликано — коммитабельно!

```
> drupal config:export
```

```
core.entity_form_display.product_entity.product_entity.default.yml  
core.entity_view_display.product_entity.product_entity.default.yml  
field.field.product_entity.product_entity.field_colors.yml  
field.field.product_entity.product_entity.field_image.yml  
field.storage.product_entity.field_colors.yml  
field.storage.product_entity.field_image.yml  
views.view.products.yml
```

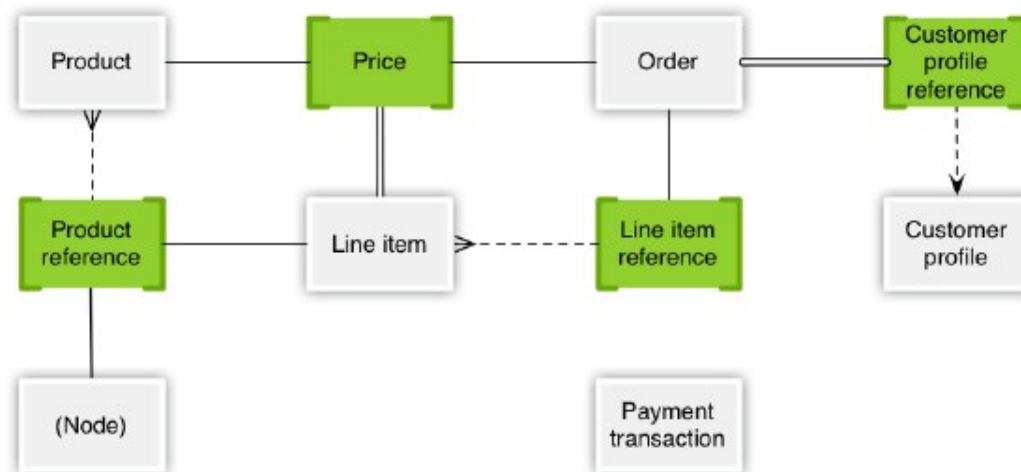


# Drupal 8

1.  Symphony components
2.  Drupal traditions
3.  Community contributions

# Drupal Commerce

Фреймворк во фреймворке 🤖



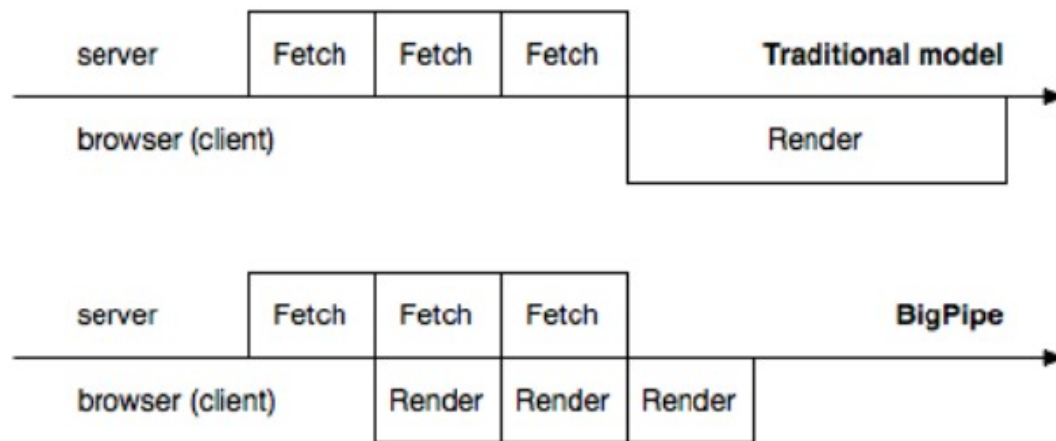
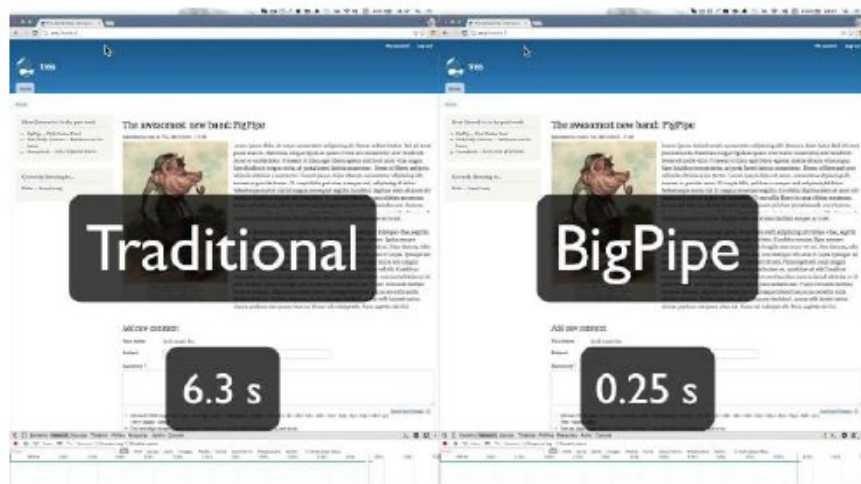
**commerce**   
**kickstart™**

**platform**   
from **commerce**  
**guys**

# Тегированный (умный) кеш

```
$build = array(  
  '#type' => 'markup',  
  '#markup' => t('Hello World!'),  
  '#cache' => [  
    'keys' => ['entity_view', 'node', 5, 'teaser'],  
    'contexts' => ['languages', 'timezone'],  
    'tags' => ["node:5", 'user:3'],  
    'max-age' => Cache::PERMANENT,  
  ],  
);
```

# Модуль BigPipe в ядре



# 115 языков и групп переводчиков на [localize.drupal.org](https://localize.drupal.org)

Language ▼	Drupal 8.4.8 core progress	Contributors
Ukrainian	<div><div></div></div>	102
French	<div><div></div></div>	568
Hungarian	<div><div></div></div>	206
Dutch	<div><div></div></div>	425
Norwegian Bokmål	<div><div></div></div>	80
Spanish	<div><div></div></div>	594
Polish	<div><div></div></div>	134
Romanian	<div><div></div></div>	86
Japanese	<div><div></div></div>	80
Russian	<div><div></div></div>	269
Italian	<div><div></div></div>	157
Danish	<div><div></div></div>	87
German	<div><div></div></div>	403
Estonian	<div><div></div></div>	33
Finnish	<div><div></div></div>	81
Catalan	<div><div></div></div>	77
Chinese, Simplified	<div><div></div></div>	231
Swedish	<div><div></div></div>	143



# Drupal — локомотив Semantic Web

Stéphane Corlosquet was responsible for the integration of  
schema.org  
into Drupal... (<http://schema.org/docs/about.html>)

Drupal 6

[https://www.drupal.org/project/  
rdf](https://www.drupal.org/project/rdf)

+ кастомный код в теме

Drupal 7

RDFa в ядре

<https://www.drupal.org/project/rdfx> - RDF UI

<https://www.drupal.org/project/schemaorg>

[https://www.drupal.org/project/commerce\\_goodrelations](https://www.drupal.org/project/commerce_goodrelations)

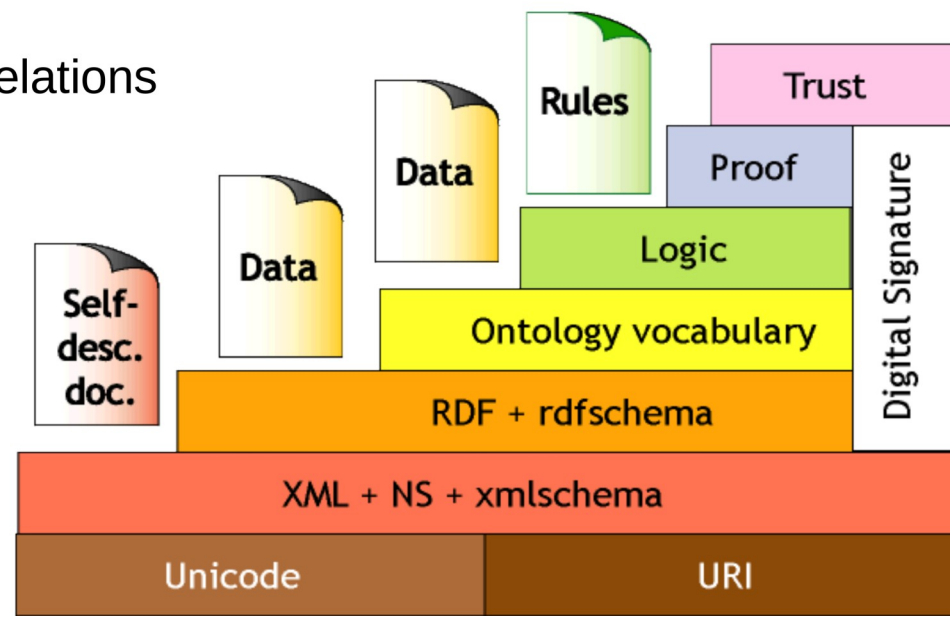
Drupal 8

RDFa/schema.org в ядре

<https://www.drupal.org/project/rdfui>

[https://www.drupal.org/project/schema\\_metatag](https://www.drupal.org/project/schema_metatag)

<https://www.drupal.org/project/ontology>



# Headless Drupal и веб-сервисы

- Доступ ко всем сущностям контента через RESTful API, (hal\_json, XML, JSON в ядре)
- Файлы и конфигурация в разработке
- JSON API (<https://www.drupal.org/project/jsonapi>)
- GraphQL (<https://www.drupal.org/project/graphql>)

# Drupal 8 RESTful сервисы ядра

- Построен на модуле Serialization: API сериализации и нормализации.
- По умолчанию JSON, в отдельном модуле JSON+HAL.
- Позволяет манипулировать сущностями контента (ноды, пользователи, комментарии, ...) и конфигурации (словари таксономии, роли пользователей, ...), модули могут добавлять свои ресурсы.
- Поддерживает GET/POST/PATCH/DELETE.
- Интегрирован с системой аутентификации Drupal: отдельные права доступа к каждому ресурсу и операции.

# Модули веб-сервисов Drupal 8

## ▼ WEB SERVICES

<input checked="" type="checkbox"/>	<b>HAL</b>	► Serializes entities using Hypertext Application Language.
<input checked="" type="checkbox"/>	<b>HTTP Basic Authentication</b>	► Provides the HTTP Basic authentication provider
<input checked="" type="checkbox"/>	<b>JSON:API</b>	► Exposes entities as a JSON:API-specification-compliant web API.
<input checked="" type="checkbox"/>	<b>RESTful Web Services</b>	► Exposes entities and other resources as RESTful web API
<input checked="" type="checkbox"/>	<b>Serialization</b>	► Provides a service for (de)serializing data to/from formats such as JSON and XML.

# Drupal 8 права доступа

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
<b>RESTful Web Services</b>			
Administer REST resource configuration	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
<b>Node</b>			
<i>Article</i> : Create new content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Basic page</i> : Create new content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Order</i> : Create new content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Article</i> : Delete any content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Basic page</i> : Delete any content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Order</i> : Delete any content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Article</i> : Delete own content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Basic page</i> : Delete own content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<i>Order</i> : Delete own content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



# Drupal 8 REST UI

Установить <https://www.drupal.org/project/restui>

или в командной строке

```
composer require drupal/restui
```

```
drush en restui
```

См. </admin/config/services/rest>

## REST resources

Here you can enable and disable available resources. Once a resource has been enabled, you can restrict its formats and authentication by clicking on its "Edit" link.

### Enabled

RESOURCE NAME	PATH	DESCRIPTION	OPERATIONS
Content	<code>/node/{node}</code> : GET, PATCH, DELETE <code>/node</code> : POST	methods: GET, POST, PATCH, DELETE formats: hal_json authentication: basic_auth	<button>Edit</button> ▼

### Disabled

RESOURCE NAME	PATH	DESCRIPTION	OPERATIONS
Action <sup>(read-only)</sup>	<code>/entity/action/{action}</code> : GET		<button>Enable</button>
Base field override <sup>(read-only)</sup>	<code>/entity/base_field_override/{base_field_override}</code> : GET		<button>Enable</button>
Block <sup>(read-only)</sup>	<code>/entity/block/{block}</code> : GET		<button>Enable</button>

# Drupal 8 получение ресурса

Включаем авторизацию через Cookies и  
Формат JSON.

См. [http://workflows.docksal/node/2?\\_format=json](http://workflows.docksal/node/2?_format=json):

```
{
  "nid": [ {
    "value": 2 } ],
  "uuid": [ {
    "value": "3253e0da-21b2-4ebd-8adf-6550d4ad9047" } ],
  "vid": [ {
    "value": 23 } ],
  "type": [ {
    "target_id": "order",
    "target_type": "node_type",
    "target_uuid": "d56df9b3-ef14-4224-ba00-a05c01432f60" } ],
  ...
  "title": [ {
    "value": "Some title prog" } ],
  "uid": [ {
    "target_id": 1,
    "target_type": "user",
    "target_uuid": "18884eb4-bae7-4eb4-bf4e-5341e3f9fbca",
    "url": "/user/1" } ],
  "created": [ {
    "value": "2019-04-04T09:50:29+00:00",
    "format": "Y-m-d\\TH:i:sP" } ],
  ...
}
```

Here you can restrict which HTTP methods should this resource support. And within each method, the available serialization formats and authentication providers.

## Granularity

Resource ▼

## Methods

- ☒ GET
- ☒ POST
- ☒ DELETE
- ☒ PATCH

## Accepted request formats

- ☒ hal\_json
- ☐ json
- ☐ xml

## Authentication providers

- ☒ basic\_auth
- ☐ cookie

Save configuration

# Views REST Export

Structure → Views → Add View

## VIEW BASIC INFORMATION

View name \*

Orders

Machine name: orders [\[Edit\]](#)

☐

Description

## VIEW SETTINGS

Show:

Content

of type:

Order

sorted by:

Newest first

## PAGE SETTINGS

☐

Create a page

## BLOCK SETTINGS

☐

Create a block

## REST EXPORT SETTINGS

☒

Provide a REST export

REST export path

orders

Save and edit

Cancel

# Views REST Export

## Displays

REST export

+ Add

Display name: REST export

### TITLE

Title: [None](#)

### FORMAT

Format: [Serializer](#) | [Settings](#)

Show: [Entity](#)

### FIELDS

The selected style or row format does not use fields.

### FILTER CRITERIA

Add



[Content: Published \(= Yes\)](#)

[Content: Content type \(= Order\)](#)

### SORT CRITERIA

Add



[Content: Authored on \(desc\)](#)

### PATH SETTINGS

Path: [/orders](#)

Authentication: [No authentication is set](#)

Access: [Permission](#) | [View published content](#)

### HEADER

The selected display type does not use header plugins

### FOOTER

The selected display type does not use footer plugins

### NO RESULTS BEHAVIOR

The selected display type does not use empty plugins

### PAGER

Items to display: [Display a specified number of items](#) | 10 items

Save

Cancel

# Views REST Export

[http://workflows.docksal/orders?\\_format=xml](http://workflows.docksal/orders?_format=xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <item key="0">
    <nid>
      <value>2</value>
    </nid>
    <uuid>
      <value>3253e0da-21b2-4ebd-8adf-
6550d4ad9047</value>
    </uuid>
    <vid>
      <value>23</value>
    </vid>
  ...
```

[http://workflows.docksal/orders?\\_format=json](http://workflows.docksal/orders?_format=json)

```
[
  {
    "nid": [
      {
        "value": 2
      }
    ],
    "uuid": [
      {
        "value": "3253e0da-21b2-4ebd-8adf-
6550d4ad9047"
      }
    ],
    "vid": [
      {
        "value": 23
      }
    ],
    ...
  }
  ...
]
```



# Создание и изменение ресурсов

```
curl --include --request POST  
  --header 'Content-Type: application/json'  
  http://workflows.docksal/node?_format=json  
  --data-binary '{"type": [{"target_id": "page"}, {"title": [{"value": "WS Title"}]}']
```

HTTP/1.1 201 Created

...

Location: http://workflows.docksal/node/3

```
{"nid":[{"value":3}], ...}
```

CSRF токен можно взять с /rest/session/token и добавить в заголовок запроса X-CSRF-Token

Авторизация через HTTP или Cookie сессии Drupal.

Примеры:

<https://www.drupal.org/docs/8/core/modules/rest/javascript-and-drupal-8-restful-web-services>

# Hypertext Application Language (HAL)

- JSON API
- Два не обязательных вспомогательных свойства:
  - (1) "\_links": ссылка на связанные ресурсы.
  - (2) "\_embedded": встроенные ресурсы.
- HAL is a simple format that gives a consistent and easy way to hyperlink between resources in your API. Adopting HAL will make your API explorable, and its documentation easily discoverable from within the API itself. In short, it will make your API easier to work with and therefore more attractive to client developers.
- [http://stateless.co/hal\\_specification.html](http://stateless.co/hal_specification.html)

GET /orders HTTP/1.1

Host: example.org

Accept: application/hal+json

HTTP/1.1 200 OK

Content-Type:  
application/hal+json

```
{
  "_links": {
    "self": { "href": "/orders" },
    "next": { "href": "/orders?
page=2" },
    "find": { "href": "/orders{?
id}", "templated": true }
  },
```

```
    "_embedded": {
      "orders": [{
        "_links": {
          "self": { "href": "/orders/123" },
          "basket": { "href": "/baskets/98712" },
          "customer": { "href": "/customers/7809" }
        },
        "total": 30.00,
        "currency": "USD",
        "status": "shipped",
      }, {
        "_links": {
          "self": { "href": "/orders/124" },
          "basket": { "href": "/baskets/97213" },
          "customer": { "href": "/customers/12369" }
        },
        "total": 20.00,
        "currency": "USD",
        "status": "processing"
      }]
    },
    "currentlyProcessing": 14,
    "shippedToday": 20
  }
}
```

# JSON API

- API с нулевой конфигурацией!
- В ядре Drupal 8.7+ (!)
- <https://jsonapi.org/>

# JSON API

GET|POST /jsonapi/node/page

PATCH|DELETE /jsonapi/node/page/{uuid}

URL: /jsonapi/{entity\_type\_id}/{bundle\_id}/{entity\_id}

Filtering: ?filter[field\_name]=value&filter[field\_other]=value

Example:

```
{
  "data": [
    {
      "type": "node--page",
      "id": "acda3d12-6e11-4b52-9777-2bc6b64262ed",
      "attributes": {
        "nid": 7,
```

...



# GRAPHQL

- Придуман Facebook
- Как JSON API но типизирован, одна точка входа, более удобный язык запросов.
- Drupal 8 contrib <https://www.drupal.org/project/graphql>
- <https://www.graphql.org/>

```
{          {  
  hero {   "hero": {  
    name    "name": "Luke Skywalker",  
    height  "height": 1.72  
  }        }  
}
```

# Еще...

- Migrate
- Inplace Edit
- Layouts
- Twig, responsive images, breakpoints

# Как поставить Drupal правильно?

# Установка composer.

```
curl -sS https://getcomposer.org/installer | php
```

```
mv composer.phar /usr/bin/composer
```

# Установка Drupal 8

# <https://github.com/drupal-composer/drupal-project>

```
composer create-project drupal-composer/drupal-project:8.x-dev some-dir --no-interaction
```

# Устанавливаем все зависимости из composer.lock или composer.json

```
composer install
```

...

# Установка модуля

```
cd some-dir
```

# Devel 1.0 или старше

```
composer require drupal/devel:~1.0
```

# Webform >=5.0+ но < 6

```
composer require drupal/webform:^5.0
```

# Captcha последней стабильной версии

```
composer require drupal/captcha
```

# Обновляем зависимости в composer.lock

```
composer update
```

```
composer install
```

# Включаем модуль

```
drush en devel
```

Или ставить через Docksal.io для локальной разработки в Docker

# Ставим Drupal 8 с с профилем «стандарт»

```
git clone https://github.com/docksal/drupal8.git drupal
```

```
cd drupal
```

```
fin init
```

# Альтернативно, ставим Drupal 8 с с профилем «Umami Demo»

```
git clone https://github.com/mrsin/boilerplate-drupal8.git drupal8
```

```
cd drupal8
```

```
fin init
```

# Рестарт серверов

```
fin start
```

# Стоп серверов

```
fin stop
```

# Установка модуля одной командой

```
fin exec composer require drupal/name_of_module
```

# Факультативный проект – корпоративный сайт на Drupal

Настроить каталог товаров с отбором по параметрам и галереей изображений в каждом товаре.

Тип нода – товар.

Параметры – словари таксономии.

Views – страницы каталога и галерея.

Webform или Drupal Commerce – заказ товара.

**Bonus:** фронтэнд или мобильное приложение с бекэндом на Drupal и связью по JSON API или GraphQL.

# Ссылки

Блог и курсы команды Drupal Coder:  
<https://drupal-coder.ru/>

Доклад Кати Маршалкиной на DevConf 2016:  
<https://devconf.ru/ru/archive/devconf2016/offer/160>