

What's New in Drupal 8

Entity Field API



What We're Talking About

- ▶ How entities are used in Drupal 7
- ▶ Improvements to the entity API
- ▶ Improvements to the field API
- ▶ Other new things that are made easier, or possible, by these improvements
- ▶ What you can do to start learning now



Drupal 7 Entities & Fields

- ▶ Introduced late in the development cycle—laid the groundwork for consistency but comes up short
- ▶ Fields and properties are handled differently
- ▶ `EntityFieldQuery` is limited in functionality
- ▶ Enhancements in contributed entity module

Everything Is an Entity

- ▶ Universal API for data CRUD
- ▶ All content is stored as entities
- ▶ Complex configuration is stored as entities
- ▶ Your data should be stored as entities

Entities

- ▶ Entities are typed objects
- ▶ `YourEntity` implements `EntityInterface`
- ▶ `EntityManager` for interacting with entities of any type
- ▶ Everything is a field

Fields

- ▶ Base fields vs. bundle fields
- ▶ Consistent API for accessing field values
- ▶ Widgets & formatters work for all fields
- ▶ Chainable, and smart about relationships

Code Examples

```
// Drupal 7
```

```
print $node->field_name[LANGUAGE_NONE][0]['value'];
```

```
// Drupal 8
```

```
print $node->field_name->value;
```

```
$author_name = $node->uid->entity->name->value;
```

Code Examples

```
// Drupal 7  
$field = field_info_field($field_name);  
$instance = field_info_instance($entity_type, $field_name, $bundle);  
// Drupal 8  
$field_definition = $entity->field_name->getFieldDefinition()
```


Code Examples

```
// Drupal 8  
$node->field_name->value = 'Hello world';  
$node->save();
```

Handlers

- ▶ Entities represent a piece of data, handlers are responsible for acting on that data
- ▶ Declared per entity type
- ▶ Use the defaults, or extend them for customizations

Handlers are used for:

- ▶ Storage
- ▶ Access control
- ▶ Rendering
- ▶ Forms
- ▶ Listing
- ▶ etc.

EntityQuery

- ▶ Improved EntityFieldQuery
- ▶ No more `propertyCondition()` vs. `fieldCondition()`, just `condition()`
- ▶ Don't query the database, extend the storage handler and add your own methods

Translations

- ▶ Native support for translation of all entity content
- ▶ Easily access translated values for an entire entity or individual fields

```
$translation = $node->getTranslation('es');
```

```
// English title
```

```
print $node->title->value;
```

```
// Spanish title
```

```
print $translation->title->value;
```

Typed Data API

- ▶ Metadata about the data stored in a field
- ▶ Predictable and consistent representation
- ▶ Makes it easier for REST, Views, Tokens, etc. to access data

What Do I Need to Know?

- ▶ Basic object-oriented PHP to use existing entities
- ▶ Drupal 8's plugin system to create new entity types
- ▶ Annotations

Recap

- ▶ Unified Entity Field API – Everything is an entity
- ▶ Improved API for accessing field data
- ▶ Full CRUD support and additional management provided by handlers
- ▶ Improvements to related tools made possible by updating the underlying architecture

