



Building custom field types

Edi Modrić (@emodric)

<https://github.com/emodric/EzscIpAddressBundle>



About me



- Edi Modrić
- Computer engineering masters degree
- Working at Netgen for 6 years
 - 5 years of eZ Publish development
 - Certified eZ Publish developer
 - eZ Tags maintainer
 - Collaborated with eZ Systems on eZ Publish 5 kernel
- @emodric
- edi@netgen.hr

Agenda



- A little bit of theory
- Coding a simple field type
- Verifying that it works



The code

```
$ cd /var/www/ezpublish/src/  
$ mkdir Netgen && cd Netgen  
$ git clone  
git://github.com/emodric/Ezsc  
IpAddressBundle.git
```



Datatype + public API = field type

- Content class = Content type
- Content object = Content
- Content class attribute = Field definition
- Content object attribute = Field
- Object attribute content = Field value
- Datatype = Field type



Field type components

- Main field type implementation
- Field type value
- Per storage converters



Field type storage

- Internal storage
 - Stores data to ezcontentobject_attribute DB table
- External storage
 - Custom database tables
 - Anything else



What about legacy stack?

- Legacy datatypes are not going anywhere
- No admin interface based on Symfony stack
- To actually use your field type, you **WILL** need legacy datatype



Field type vs. datatype

- No difference in functionality
- Advantages over datatypes
 - Clear separation of concerns
 - No POST variables
 - No SQL
 - No file uploads
 - More readable and understandable code