

TYPOLight developer workshop

The goal of the workshop was to discuss some very general programming problems that do not relate to TYPOLight in particular but basically to every web-based application.

1. Best practice for "speaking URLs"

How to generate search engine friendly URLs that are not nested too deep and how to prevent Google from indexing the same page multiple times (duplicate content) if e.g. only the date of the mini-calendar changes?

Using a suffix

Using an .html suffix reduces the nesting level, which is good for the page rank.

http://domain.de/alias/key/value ← seen as index.html in folder value (4th level)

http://domain.de/alias/key/value.html ← seen as value.html in folder key (3rd level)

Using *key:value* pairs

Using *key:value* pairs instead of *key/value* pairs reduces the nesting level and allows you to add parameters without having to worry about their order.

http://domain.de/alias/key/value.html ← seen as value.html in folder key (3rd level)

http://domain.de/alias/key:value.html ← seen as key:value.html in folder alias (2nd level)

2. Best practice for multi-language websites

How to add the language information to an URL so Google can index all languages and you can still use flexible aliases or even the same alias for one page in multiple languages?

No "one size fits all" solution

- Relying on the browser language only would prevent Google from indexing anything that is not english
- Adding an alias prefix (e.g. /en/alias.html) increases the nesting level
- Adding a default URL suffix (e.g. alias.en.html) can be a problem if existing URLs shall be preserved

Possible solution

We could provide an option to choose the position of the language code. It can either be added before the alias, before the URL suffix or not at all. The solution should follow the "one alias per language" principle, however, you can use the same alias multiple times as long as the language tag is part of the URL.

3. InnoDB vs. MyISAM

For using hierarchical categories, you should use nested sets. Since nested sets are very sensitive while they are written to the database, transactions should be used to ensure the referential integrity. However, MyISAM tables (the default type of table in MySQL) do not support transactions before version 5.3.

Can we make InnoDB a requirement?

- Not all providers support InnoDB tables
- It is not sure if InnoDB remains a part of the MySQL project

Possible solution

- Use transactions and InnoDB if available
- Fallback solution for MyISAM tables: do not use transactions
- Create driver methods to support specific features like the MySQL full-text search

4. Meta information for images and files

How to save meta information for images and files in the database without having to save the image itself in the database as well? How to ensure referential integrity if a client renames or moves a file via FTP?

Possible solution

Use `md5_file()` to create a unique hash that can be stored in the database instead of the file. Create a maintenance job that checks the file system for moved or renamed (eventually even deleted) files.

Saved by the bell

Unfortunately, we ran out of time pretty quickly so we could not discuss all the topics on the list. The following two items were left:

- How to store recurring events in the database?
- MooTools vs. jQuery