

User Application

Simple Application
to do one thing

Goals:

- Easy to use
- Quick to start

Re-use:

- HTTP stack

- JSON

Custom route:

```
from trytond.application import app
```

```
@app.route('/hello', methods=['GET'])
```

```
def hello(request):
```

```
    return "Hello world"
```

No user/password

Application Key

- created by the application
- validated by the user (preferences)

Create Application Key

POST '<database>/user/application/'

data: {

 "user": <user>,

 "application": <name>

}

returns: <key>

Delete Application Key

```
DELETE '<database>/user/application/'
```

```
data: {
```

```
  "user": <user>,
```

```
  "key": <key>,
```

```
  "application": <name>
```

```
}
```

Some helpers

```
from trytond.protocols.wrappers import *
```

```
@app.route('/<database>/hello')
@with_pool
def hello(request, pool):
    ...
```

```
@app.route('/<database/hello')
@with_pool
@with_transaction()
def hello(request, pool):
    ...
```

`with_transaction(readonly=None)`

POST, PUT, DELETE, PATCH: not readonly

others: readonly

```
@app.route('/<database>/hello')
@with_pool
@with_transaction()
@user_application('myapp')
def hello(request, pool):
    ...
```

user_application(name)

- Check key
in Authorization (type: bearer)
- Set user to Transaction
- Activate check access

```
user_application(name, json=True)
```

Force 'application/json' response

Example

Timesheet/Chronos