

Tapestry5HowToDomainLevelLocalization

Inside [AppModule.java](#):

```
@Inject
private RequestGlobals requestGlobals;

@Inject
private ThreadLocale threadLocale;

public void contributeRequestHandler(OrderedConfiguration<RequestFilter> configuration,
    @InjectService("TimingFilter")
    RequestFilter filter,
    ThreadLocale locale)
{
    // Each contribution to an ordered configuration has a name, When necessary, you may
    // set constraints to precisely control the invocation order of the contributed filter
    // within the pipeline.

    Map<String, Locale> serverNameMapping = new HashMap<String,Locale>();

    serverNameMapping.put("www.domain.com", Locale.ENGLISH);
    serverNameMapping.put("www.domain.de", Locale.GERMAN);
    serverNameMapping.put("www.domain.fr", Locale.FRENCH);

    configuration.add("Timing", filter);
    configuration.add("HostBasedLocalization", new HostBasedLocaleFilter(threadLocale, Locale.GERMAN,
serverNameMapping), "after:Localization");
}
```

[HostBasedLocaleFilter.java](#)

```

import java.io.IOException;
import java.util.Locale;
import java.util.Map;

import org.apache.tapestry.ioc.services.ThreadLocale;
import org.apache.tapestry.services.Request;
import org.apache.tapestry.services.RequestFilter;
import org.apache.tapestry.services.RequestHandler;
import org.apache.tapestry.services.Response;

public class HostBasedLocaleFilter implements RequestFilter {

    private Map<String, Locale> serverNameMapping;

    private Locale                defaultLocale;

    private ThreadLocale          threadLocale;

    public HostBasedLocaleFilter(final ThreadLocale threadLocale, final Locale defaultLocale, final
Map<String, Locale> serverNameMapping) {
        this.serverNameMapping = serverNameMapping;
        this.defaultLocale = defaultLocale;
        this.threadLocale = threadLocale;
    }

    public boolean service(Request request, Response response, RequestHandler handler) throws IOException {
        String servername = request.getParameter("Host");
        System.out.println(servername);
        Locale locale = this.serverNameMapping.get(servername);
        if (locale == null) {
            locale = this.defaultLocale;
        }
        threadLocale.setLocale(locale);

        return handler.service(request, response);
    }
}

```