Browser Runtime Programming Using TypeScript
Latest In-Browser Storage, Security, Networking, Formats and Device Info

The programming environment inside the modern web browser has significantly matured and now offers a rich and diverse range of capabilities, some at the UI level and some at the underlying runtime level. Many of the newer features are currently not being fully exploited by web developers. This course aims to change that by exploring in depth the non-UI aspects of browser programming, using the TypeScript language for all demos and lab exercises.

The runtime programming APIs in a modern browser now rival what modern OSes offer. A web app has many natural advantages over mobile apps (here's four: the power of the URI, works everywhere, cloud-friendly, immediate app updates). In areas such as sandboxed file access, security, networking, data formats and device info, a modern browser offers the web application developer a comprehensive selection of functionality that when used correctly can easily compete with what is available for native mobile apps.

IMPORTANT: This course does not cover parallel (web worker) & asynchronous programming – we offer a separate full course that covers this in detail.

Target Audience

Web developers wishing to fully leverage the runtime (non-UI) capabilities of modern web browsers.

Prerequisites

Good experience of web development, including HTML 5.2.

Knowledge of the TypeScript programming language.

Contents of One-Day Training Course

Modern Web Platform

Tour of the modern web platform
Major features and which browsers implement them (caniuse.com website)
Dynamically detecting in code available browser capabilities and optimizing app

File API

File Sandboxing
Accessing and updating files
File metadata and raw blob content

Indexed DB

A B-tree like persistence mechanism with powerful indexing that can serve as basis for in-browser client database
Comparison with other storage options

Overview of Browser Security

TLS and SSL
Input validation
Output encoding
Cookies and security

Content Security Policy

CSP feature tour
Directives
Policy definition
Integration with other specifications

Strict Transport Security

"Defines a mechanism enabling web sites to declare themselves accessible only via secure connections" - RFC

Web Cryptography

Running cryptographic algorithms inside a browser (AES, RSA, HMAC, SHA, etc.)
A W3C recommendation from Jan 2017

HTTP/2

Latest generation of HTTP protocol
Binary, multiplexed, full duplex, priority
Server push
New app architectural possibilities

Server Sent Events

How it works
EventSource API
Event streams
Event handlers

CORS

Cross-Origin Resource Sharing
Role of origin in HTTP protocol
cross-origin access
CORS preflight request and CORS request

Fetch

A significantly improved replacement for XmlHttpRequest
"The Fetch standard defines requests, responses, and the process that binds them: fetching." (WHATWG)

Formats

Brotli Compressed Data Format
Data URIs using base64 encoding
data-* attributes

Device Info

Accessing device details
GeoLocation
Battery status

Project

Bringing together the ideas covered in this course to design and build an extensible modular system platform

CLIPCODE™

http://www.clipcode.net/training

To arrange an on-site presentation anywhere in Europe, please email training@clipcode.com