

Introduction to Git

อ. ประจักษ์ ปิยะวงศ์วิศาล

Pratch Piyawongwisal

Today

- Proposal v2
- Recap – Git
- Git
- Ionic (ep #5-6)
- Assignment 3 – Ionic/Git

Due: Assignment 2

- **Project Proposal v2** (ให้เขียนโดยใช้ HTML)
 - ชื่อโครงการ, รายชื่อสมาชิก
 - เป้าหมายของโครงการ จะพัฒนาซอฟต์แวร์อะไร (สั้นๆ ไม่เน้นวิธีการ/เครื่องมือ)
 - ลูกค้า/ผู้ใช้เป็นใคร
 - ซอฟต์แวร์ที่พัฒนาตอบโจทย์ความต้องการของผู้ใช้อย่างไร ช่วยแก้ปัญหาอะไรได้บ้าง
 - รายการฟีเจอร์ความสามารถ
 - เครื่องมือที่จะใช้
- **Ionic Hello World, Navigating between Pages**
 - watch สอน Ionic 3, Ep #5-6
 - https://www.youtube.com/watch?v=vSCoY_ghW-4&list=PLiPfp4Ph3gBp_F7v0bqxmfvO4r0ceDA9&index=1

Git

Recap: Version Control - Git

- VCS (Version Control System) เป็นเครื่องมือที่ทีมนักพัฒนาใช้ในการแชร์ codebase
- Git เป็น VCS แบบ Distributed (ไม่มี Server กลาง)
 - พัฒนาโดย Linus Torvalds ผู้สร้าง Linux
 - open source ใช้กันเป็นที่แพร่หลาย
 - Git repo เป็นที่เก็บ codebase รวมถึงประวัติการ commit ไว้ทั้งหมดในลักษณะกราฟ
 - ใช้คำสั่งบน command line
- GitHub เป็นเว็บสำหรับโฮสต์ Git repo ไม่จำกัด ฟรี (แบบ public เท่านั้น)

Git Basics

- Download Git
 - <https://git-scm.com/downloads>
- Create a GitHub account
 - <https://github.com/join>
- Watch Harvard CS50 Git Seminar
 - https://www.youtube.com/watch?v=MJUI4wbEm_A

Git - Basic Commands

- `git clone <url>` หรือ `git init`
- `git add <filename>`
- `git commit -m "message"`
- `git status`
- `git push`
- `git pull` (อาจเกิด `conflict`)
- `git log`
- `git reset`

Git - Branching Commands

- `git branch`
- `git merge` (อาจเกิด `conflict`)
- `git checkout`

Git - Advanced Commands

- `git rebase`

Git Lab – Part 1: Local repo

Getting Started – first commit

- สร้างโฟลเดอร์ชื่อ **project** แล้วเปิดใน **command prompt**
- สร้าง **git repo** เปล่าๆ ด้วยคำสั่ง
 - `git init`
- สร้าง **readme.txt** แล้วเช็คสถานะด้วยคำสั่ง
 - `git status`
- **staging** เพิ่มไฟล์ **readme** ที่จะ **commit** ด้วยคำสั่ง
 - `git add readme.txt`
- ทำการ **commit** (สร้าง **checkpoint**) ด้วยคำสั่ง
 - `git commit -m "Added readme.txt"`

Getting Started – first commit

- ลองแก้ไข `readme.txt` แล้วเช็คสถานะ
 - `git status`
- หากจะ `commit` ไฟล์ที่ถูกแก้ไข จะต้อง `add` ไฟล์นั้นใหม่ก่อน
 - `git add readme.txt`
- ทำการ `Commit` ไปยัง `repo` ได้
 - `git commit -m "Edited readme.txt"`
- ตรวจสอบความเรียบร้อย
 - `git status`

Visualizing commits with GitHub Desktop

- Download
 - <https://desktop.github.com/>
- ลองเปิด **repo** ของเราด้วยโปรแกรม **GitHub Desktop** เพื่อให้เห็นภาพ

Branching - การสร้างกิ่งงานใหม่

- Create a new **alpha** branch
 - `git branch alpha`
- Switch to **alpha** branch
 - `git checkout alpha`
- Create "test_login.html" and commit
- Switch back to **master** branch
 - `git checkout master`
- Merge **alpha** into **master**
 - `git merge alpha`

Tips

- หากต้องการ `add` ทุกไฟล์ที่แก้ไข สามารถใช้คำสั่ง
 - `git add .`
- หากต้องการ `commit` เลยโดยข้ามขั้นการ `add` สามารถใช้ `option -a`
 - `git commit -a -m "updated index.html"`
- สามารถตั้งชื่อ/อีเมลตนเองด้วยคำสั่ง `git config`

Resolving merge conflicts

Conflict เกิดจากการแก้ไขไฟล์เดียวกันไปคนละทิศคนละทาง

- ไปที่ **branch master**
- เพิ่มข้อความ "change1 เว้นบรรทัด change2" ในไฟล์ `readme.txt`
- ทำการ **commit**
- กลับไป **branch alpha**
- ใน `readme.txt` เปลี่ยน `change2` เป็น `change3` แล้ว **commit**
- สังเกต **commit graph** ใน **GitHub Desktop**
 - จะพบว่าเกิดโค้ด 2 เวอร์ชันที่ต่างกันระหว่าง **master** กับ **alpha**

Resolving merge conflicts

- ลองพยายาม `merge master` เข้ากับ `alpha` **conflict!**
 - เนื่องจากมีการแก้ไขไฟล์เดียวกันไปคนละทาง
- แก้ `conflict` โดยเปลี่ยนไฟล์ `readme.txt` ให้เป็นแบบที่ต้องการ
 - `<< HEAD` หมายถึงส่วนของ `branch` ปัจจุบัน
 - `>> master` หมายถึงส่วนของ `branch master` ที่จะ `merge` เข้ามา
- เมื่อแก้ `conflict` ด้วยมือเรียบร้อยแล้ว `commit`
 - `commit -a -m "merge from master"`
- เสร็จสิ้นการ `merge`

Git Lab – Part 2: Remote repo

Creating a GitHub repo

- Go to GitHub
- Click + -> New Repository
- Name the repo "Test Project"
- Check "Initialize this repo with a README"
- Click "Create Repository"

Pushing project files to GitHub repo

- Go to project folder (where we git init'd before):
 - `git remote add origin <.git URL>`
 - `git pull origin master --allow-unrelated-histories`
 - `git push origin master`
 - `git push origin alpha`

Clone/Fork the repo to PC

- To clone entire repo to local machine:
 - `git clone <URL to .git>`
- To give push permission to friends/coworkers:
 - Go to GitHub repo page
 - Go to Settings -> Collaborators
 - Add friends

Visualizing Commits

- On GitHub repo page
 - go to Insight -> Network

Ionic

Ionic – Project Structure

- src
 - pages
 - Home
 - Contact
 - About
 - Page files
 - TypeScript (.ts)
 - HTML5+Angular (.html)
 - Stylesheet (.scss)
- | | |
|--------------------|------------------------|
| controller: | control view with code |
| view: | layout and positioning |
| styling: | custom styles |

Ionic - Tutorial Ep 5-6

- Creating a new page
 - `ionic g page <page_name>`
- Add new page to `app.module.ts` in `@NgModule`
 - `declarations`
 - `entryComponents`
- Change home page in `app.component.ts`
- Add a button to a page
- Add `onclick()` to navigate to another page
 - `this.navCtrl.push(<page_name>)`

Assignment 3

- Create GitHub repo for Team Project
 - Put all documents in: doc/
 - Add Ionic App folder to root directory
- งานเดี่ยว: ศึกษาวิธีเชื่อมต่อ Ionic App กับฐานข้อมูล Firebase
 - <http://nextflow.in.th/2017/build-mobile-app-and-progressive-web-app-with-firebase-database-angular-ionic/>