Team Project

Best reviewer 양성과정

이전의 팀프로젝트

- 사원 관리를 위한 Database를 생성하고 관리하는 프로그램을 구현을 진행했습니다.

사원 번호	성명	경력개발단계	전화번호	생년월일
17843022	KFI SEO	CL3	010-4837-6716	19810630
16982964	EUNXMFD OH	CL1	010-1527-7302	19880716
17530481	CHHDZ LEE	CL2	010-2338-2314	19940929

[사원 정보]

- **CRUD (create, read, update, delete) 구현**
- **새로운 field 를 추가**
- **성명 field 분리(Last name, First name)**
- **Read 복합 조건 추가**
  - 비교 조건(greater, lesser)을 이용해 Database 를 검색한다
  - and/or 복합 조건을 이용해 Database 를 검색한다

목표는 같지만 다른 프로젝트

```
115  int pre_Func_name_search(int option3, int option_ao) // option3 ~
116  {
117          if (option_ao) // 옵션 ao 있을 때
118          {
119                  if (option_ao == 1) // AND
120                  {
121                          if (option3) // 옵션 3 있을 때
122                          {
123                                  if (option3 == 2 || option3 == 4)
124                                  {
125                                          for (int i = 0; i < temp_l
126                                          {
127                                                  if (0 > _strcmp(va
128                                                  {
129                                                          temp_list[
130                                                          i--;
```

목표는 같지만 다른 프로젝트

```
304  int pre_Func_name_f_search(int option3, int option_ao) // option3
305  {
306          if (option_ao) // 옵션 ao 있을 때
307          {
308                  if (option_ao == 1) // AND
309                  {
310                          if (option3) // 옵션 3 있을 때
311                          {
312                                  if (option3 == 2 || option3 == 4)
313                                  {
314                                          for (int i = 0; i < temp_l
315                                          {
316                                                  if (0 > _strcmp(va
317                                                  {
```

목표는 같지만 프로젝트

```
359         if (op == 1 || op == 2) {                       // > or >=
360             //if (op == 1) iter_idx++;                      // great
361             if (is_string) {                                  // tar
362                 data_node* iter_node;
363                 while (iter != NULL) {
364                     while (iter_idx < 7 && iter->leaf.
365                         iter_node = iter->leaf.dat
366                         while (iter_node != NULL)
367                             if (res->res_cnt++
368                             //cout << iter_nod
369                             if (op2 == 2 || op
370                             iter_node = iter_n
371                     }
372                 }
```

팀프로젝트 | 개요

■ 목적

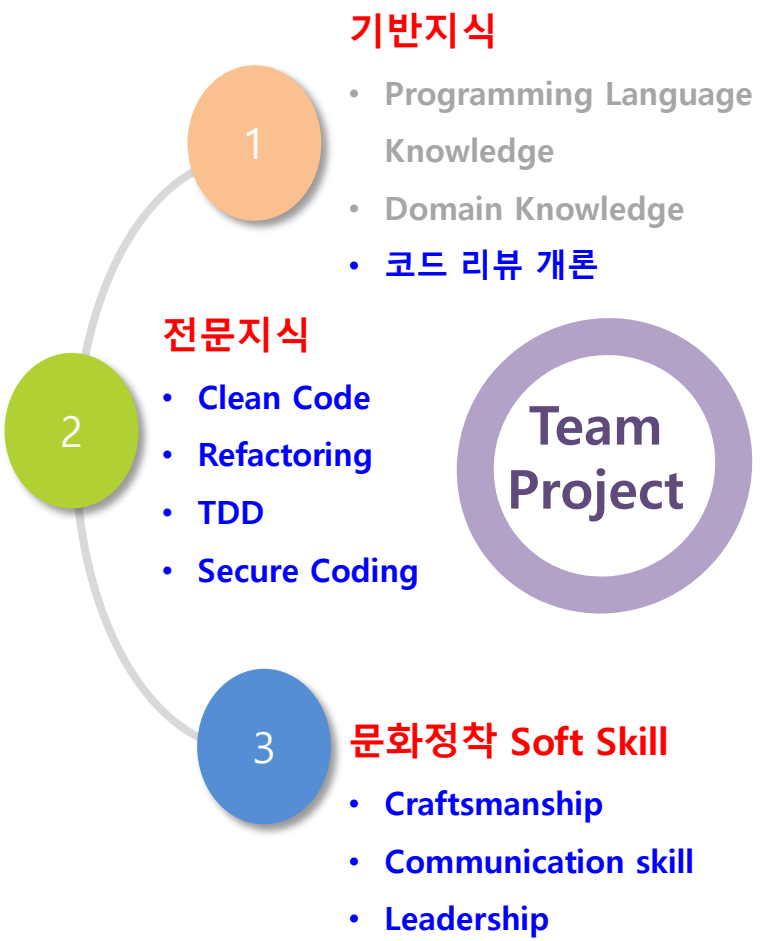
- 교육 내용의 주도적 활용 및 팀원간 소통 능력 배양

■ 특징

- 현업 프로젝트와 유사한 개발 방식
- Clean Code, TDD, Refactoring, Secure Coding 등의 지식을 활용한 요구사항 설계

■ 진행방식

- 1팀 4~5인
- 팀 내 code review 필수



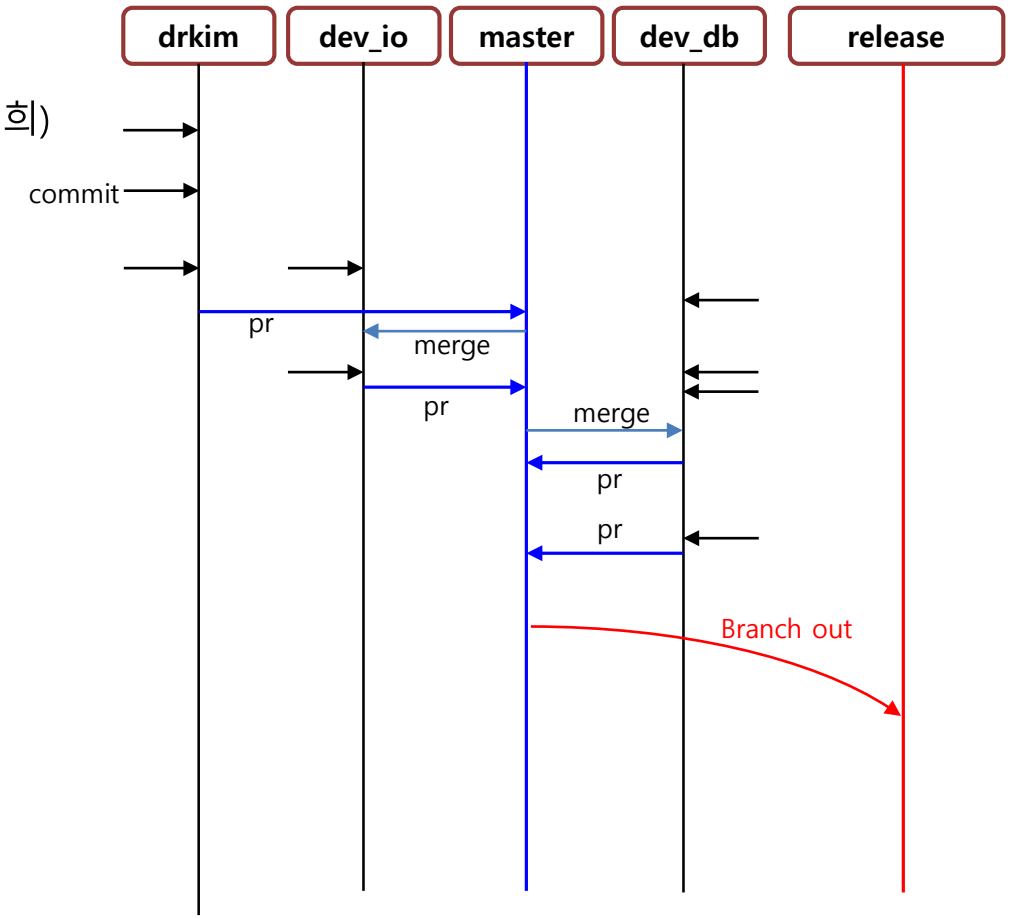
팀프로젝트 | 일정

- 1 일차
  - Base code 및 요구사항 전달
  - Base code 분석 및 요구사항 분배
  - 그라운드 룰
- 2 ~ 4 일차
  - 요구사항 구현

    회고(4일차)
- 5 일차
  - 발표자료 정리
  - 팀별 발표

팀프로젝트 | 진행방식

- Organization 내 팀 프로젝트 진행을 위한 Repository 생성
  - Repository : TeamProject_팀명 (팀명은 자유)
  - Description : 조, 참여자 이름 필수(ex: 1팀, 김박사, 김철, 윤박사, 윤영희)
- Master branch merge 시 Review 필수
  - 개발 branch -> master branch PR & Review
  - Offline review 도 Online 기록 권장
    - github 내 Issues, Projects, Wiki 메뉴 또는 md 파일 사용
  - 단, Vote up 의 개수와 같은 세부사항은 팀에서 자율 결정
- TDD Practice 필수
- release branch 를 만들어 최종 제출
- Commit 별 code change 50 line 이하 권장
  - 여러 개의 commit 을 하나의 PR 로 작성 가능



[브랜치 구조도 예시]

팀프로젝트 | 평가방식

■ 요구사항 평가

- 주어진 요구사항을 모두 만족 해야 함

■ S/W 품질 평가

- Production Code 와 Test Code 의 가독성 및 유지보수성
- Test Code 의 적절성 및 Code Coverage (90% 이상)
- Commit & Review 평가
- Review 의 적절성 (Clean Code, Refactoring, TDD, Secure Coding 측면)
- Communication Manner
- Commit 의 규모 및 Test code 동반 여부

■ 상세 평가 방법

- 동작 정확도 확인
- 개인별 S/W 품질 평가
  - PR Inspection : Class/method 크기, 코드 복잡도, Test code 여부 확인
  - Review Inspection : 코드 품질 향상 여부, Communication skill
- 팀별 발표 상호 평가
  - PR Commit 의 코드 품질 자체와 품질 향상에 영향을 미치는 리뷰에 대해 실례를 들어 잘 설명한 팀은?
- 발표자에 대한 가점

팀프로젝트 | Free Rider Problem

master

Commits on Sep 17, 2021

Add files via upload
committed 13 days ago — a72a650

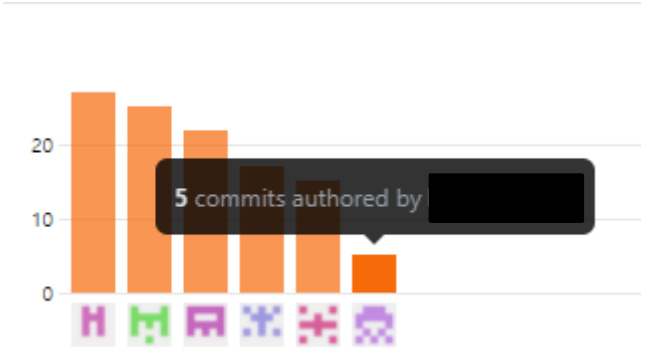
Add files via upload
committed 13 days ago — a373097

Add files via upload
committed 13 days ago

Add files via upload
committed 13 days ago

Add files via upload
committed 13 days ago

Newer | Older



master

Commits on Sep 16, 2021

EmployeeDao 기본 기능 및 TC 적용 (#36) ...
committed 14 days ago ✓ — 7338029

Commits on Sep 14, 2021

요구사항 스토리로 작성 (#9) ...
committed 16 days ago ✓ — 9784d79

Newer | Older

팀프로젝트 요구사항

Best reviewer 양성과정

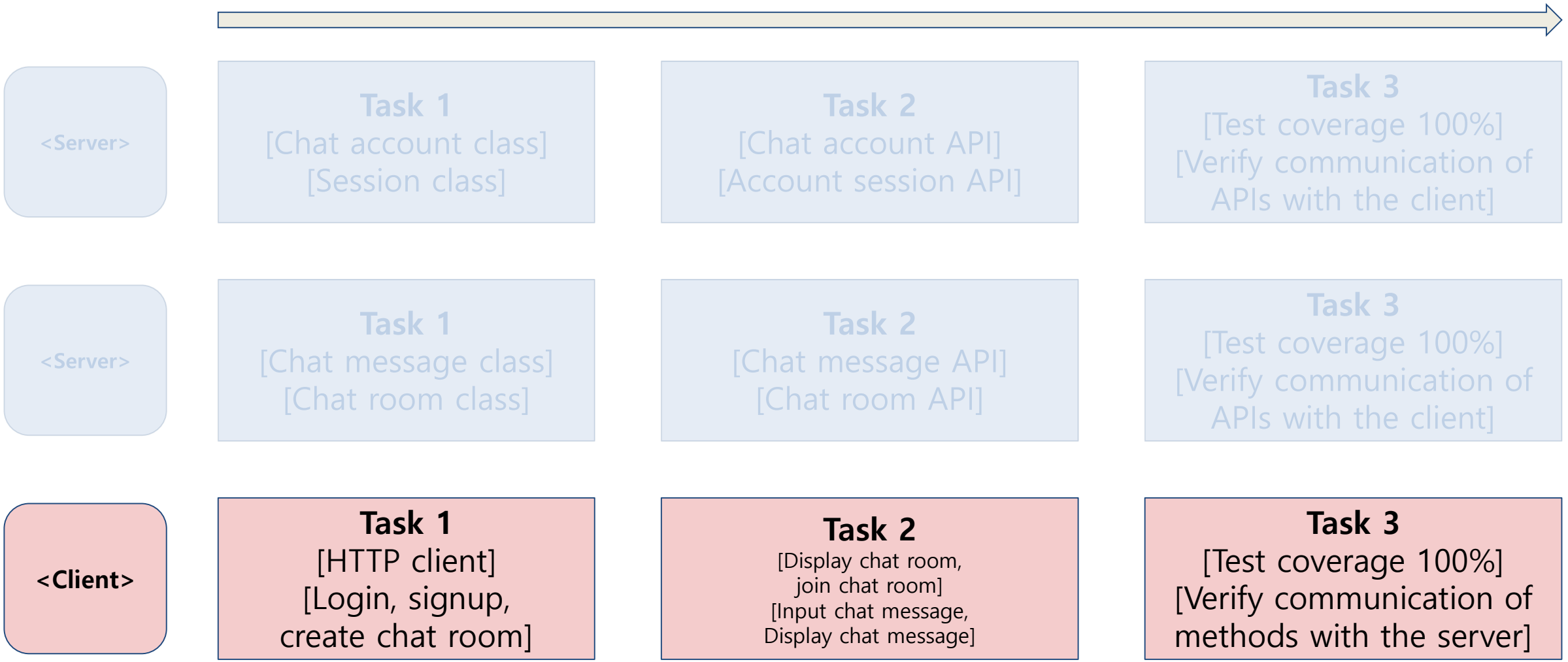
팀프로젝트 | 목표

- Code Style 사용하여 코드 가독성을 고려한 개발능력 연습 {Google C++ or etc}
- test code를 작성하는 Test-Driven Development (TDD) 능력 연습
- REST API 를 사용하여 HTTP 통신 관리 능력 학습
- Version Control System (VCS) 를 사용하여 팀 단위 공유 소스코드 관리 능력 및 커뮤니케이션 능력

팀프로젝트 | 요구사항

- RestAPI 를 이용한 Chat client 프로그램을 제작한다.
- Server 프로그램은 제공되며, 자체적으로 Server 를 실행하여 테스트 할 수 있다.
- ~~공식 Server 에 접속하여,~~
  ~~- 팀명의 아이디를 만든후,~~
  ~~- 팀명의 room을 만들어,~~
  ~~"1팀 팀프로젝트 완료" 메시지를 보내고,~~
  ~~- Room "1" 에 "N team Complete" 메시지를 보내는 것으로~~
  ~~기능이 정상적으로 동작함을 확인한다.~~
  ~~(온라인 과정으로 변경)~~

팀프로젝트 | The development plan



각 Task 에서 client code 작성과 동시에 test code 작성

팀프로젝트 | Task 상세 설명(서버측 : 이미 구현 완료)

[Chat account class]

File database 를 이용해 account 정보 저장, 읽기 기능 작성

account 생성, 조회, 저장 기능 작성

[Session class]

Session 생성, 삭제, 갱신, 조회 기능 작성
Session이 일정시간동안 사용되지 않으면 삭제되는 기능 작성

[Chat account API]

Account class와 session class를 사용하여 Login, singup, logout 기능을 제공하는 Rest API 작성

Client와 password 를 주고 받을 때 hash function과 nonce를 이용한 암호화 기능 작성

[Account session API]

Server 에서 주어진 session 이 유효한 session 인지 확인하는 기능 작성

Server의 Rest API 사용 시 세션 지속시간 갱신 기능 작성

[Test coverage 100%]

Id 중복 검사, Id에 file database의 delimiter 포함 여부 확인 등 다양한 test case 작성

[Verify communication of methods with the server]

client method가 server API와 통신이 문제 없이 하는지 교차 검증

팀프로젝트 | Task 상세 설명(서버측 : 이미 구현 완료)

[Chat message class]

Chat message 정보 관리 class 작성, message 저장 및 조회 기능 제공

File database에 chat message 저장, 읽기 기능 작성

[Chat room class]

Chat room 정보 관리 class 작성, chat room 저장 및 조회 기능 제공

File database에 chat room 저장, 읽기 기능 작성

[Chat message API]

Chat message 저장, 조회 기능 제공하는 Rest API 작성

[Chat room API]

Chat room 생성, 조회 기능 제공하는 Rest API 작성

[Test coverage 100%]

Chat message 와 chat room에 file database의 delimiter 포함 여부 확인, chat room 중복 검사 등 다양한 test case 작성

[Verify communication of methods with the server]

client method가 server API와 통신이 문제 없이 하는지 검증

팀프로젝트 | Task 상세 설명(클라이언트측)

[HTTP client]

- Chat server로 HTTP 를 사용하여 Rest API service 요청을 할 수 있는 기능 작성

[Client methods: Login, signup, create chat room]

- Login, signup, create chat room method 구현
- Server로 HTTP 요청을 하여 각 함수의 기능 작성
- Server와 password 를 주고 받을 때 hash function과 nonce를 이용한 암호화 기능 작성

[Client methods: Display chat room, join chat room]

- 존재하는 chat room 목록 보여주기, chat room 입장 기능 구현
- Server로 HTTP 요청을 하여 각 함수의 기능 작성

[Client methods: Input chat message, display chat message]

- Chat room에서 user 의 chat message 입력을 서버로 전송 기능 구현
- 실 시간으로 server의 chat message를 polling하여 chat room에 display 기능 구현

[Test coverage 100%]

- 놓치기 쉬운 예외 case 까지 test
- Characteristic test외 approval test 이용한 integration test

[Verify communication of methods with the server]

- Server에 구현된 API가 의도한대로 client와 통신하는지 검증

팀프로젝트 | 세부요구사항 | signup, login, logout

Signup

- POST /chat/account
- Body : { " id " :"{ID Name} " , " password " :"{암호}"}
- Prohobited charter : id ["," , "|"], password [","]
- Response
  - 200 OK : 정상
  - 400 Bad Request : Account information absence, prohibited character at ID/password, duplicated id, Account write error in file DB

Signup password 생성 규칙

- Body의 password 는 사용자 입력 암호를 직접 전송하지 않음
- Hash(password) 를 전송하고 서버가 저장함
  - Hash 코드 생성 규칙은 다음 페이지 참고

- ex) hash("testtest") -> 541813904

팀프로젝트 | 세부요구사항 | signup, login, logout

- Login
  - POST /chat/login
  - Body : { " id " :"{ID Name} " , " nonce " :"{NONCE}", " password " :"{PASSWORD}"}
  - Response
    - 200 OK : 정상 => json 포함 { " session_id " : " MkUdmbzNVZE678CgRV16lL1z5lyGTjal " }
    - 400 Bad Request : Account information absence(id, nonce, password가 없는 경우), ID not exist, Password error
- Login nonce, password 생성 규칙
  - nonce : random key
  - Password : hash(hash (input password) + nonce)

Server Password 확인규칙

Hash (savedPassword + nonce) == password

팀프로젝트 | hash

- 동일 Hash code 사용
- hash(“testtest”) -> 541813904

```
#include <iostream>
#include <string>
long long hash(const char *str)
{
	long long hash = 5381;
	int c;
	while (c = *str++)
	{
		hash = (((hash << 5) + hash) + c) % 1000000007;
	}
	return hash % 1000000007;
}
int main()
{
	std::string test = "testtest";

	std::cout << hash(test.c_str());
}
```

팀프로젝트 | 세부요구사항 | signup, login, logout

Logout

- DELETE /chat/session?session_id={Session ID}
- Response
  - 200 OK : 정상
  - 403 Forbidden : Not valid session id

팀프로젝트 | 세부요구사항 | room_create, room_list

room_create

- POST /chat/chatroom

- Body

  {"chat_room":"{Room Name}","session_id":"{Session ID}"}

- Response

  BadRequest : Account information absence, empty body(이외 모든 경우)

  Forbidden : Not a valid session ID

room_list

- GET /chat/chatroom?session_id={Session ID}

- Response

  [{"room":"1"},{"room":"2"},{"room":"3"},{"room":"aaa"}]

- join(join하는 Room Name의 message를 주기적으로 Receive)

  - GET /chat/chatmessage?session_id={Session ID}&chat_room={Room Name}

  - Response

    [{"date":1583135236,"message":"hello","room":“room_name","user_id":“reviewer"},

    {"date":1602568577,"message":"test","room":“room_name","user_id":"test"}]

  - BadRequest : "Chat room information missing“

  - BadRequest : "There are no chat rooms: “

  - Fobidden : "Not a valid session ID"

- Message send

  - POST /chat/chatmessage

  - Body

    {"chat_message":“{message}","chat_room":“{Room name}","session_id":”{Session ID}”}

  - Response

    - BadRequest : "Account information absence“

    - BasRequest : “Prohibited char in the message, room, or date”

    - Internal Error : “Can't find user ID from given session ID”

    - Fobidden : "Not a valid session ID"

Message Receive

- GET /chat/chatmessage?session_id={Session ID}&chat_room={room 이름}

- Response

  [{"date":1583135236,"message":"hello","room":"2","user_id":"samsung"},

  {"date":1602568577,"message":"test","room":"2","user_id":"test"}]

- BadRequest : "Chat room information missing“
- BadRequest : "There are no chat rooms: “
- Fobidden : "Not a valid session ID"

팀프로젝트 | 몇가지의 넘어야 할 고비

- http request 는 어떻게 할 것인가?
  - cpprestsdk : https://github.com/microsoft/cpprestsdk
  - curl : https://curl.se/
  - header 를 추가하는 방법은? SSL 관련 옵션을 추가하는 방법은?
- login 을 위한 nonce와 hashcode 는 어떻게 동작할 것인가?
  - 해시값 (계산식을 통해 산출될 역산이 불가능한 입력데이터에 매칭된 값)
  - 난스(Number used once; 한 번 쓰이고 버려지는 숫자)
- join은 어떻게 처리 할 것 인가? (optional)
  - 동일한 room 에서 실시간으로 message를 받기
  - Thread? 또 다른 방법?? Thread 로 구성된 로직에 대한 Test는?
- 기술적인 요소보다 더 중요한 것은 비즈니스코드와 테스트코드의 품질을 어떻게 유지할 것인가?
  - 동료들과 코드에 대한 관심과 지식교류 활동에 대한 의미를 어떻게 찾을 것인가?

팀프로젝트 | RestAPI Test tool

- RestAPI Test 를 위한 툴

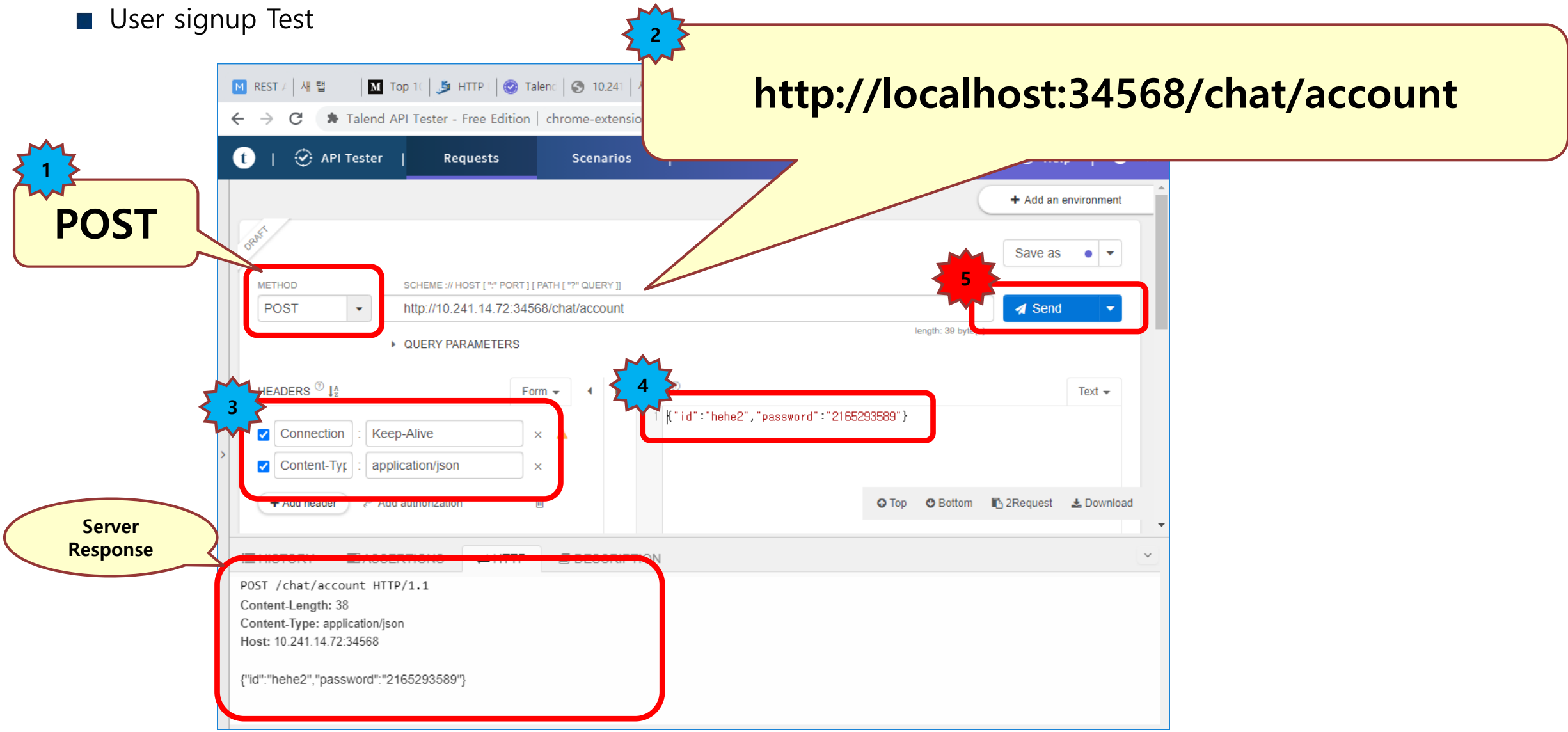
  - SaapUI, POSTMAN, Katalon Studio, 등등

- Chrome plugin 인 Talend API Tester

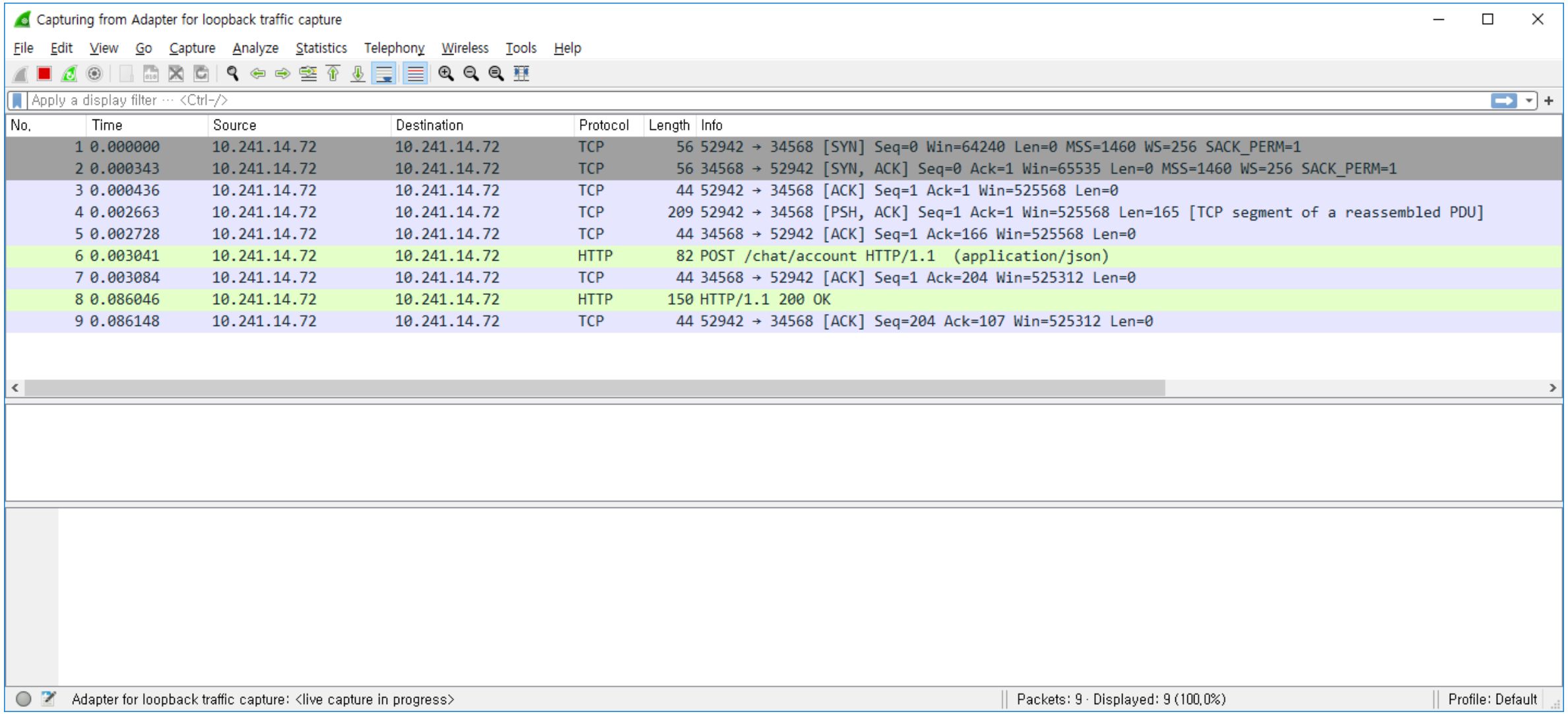
  - Chrome 웹 스토어(https://chrome.google.com/webstore/category/extensions?hl=ko) 에서 설치

팀프로젝트 | Talend API Tester(chrome web app)

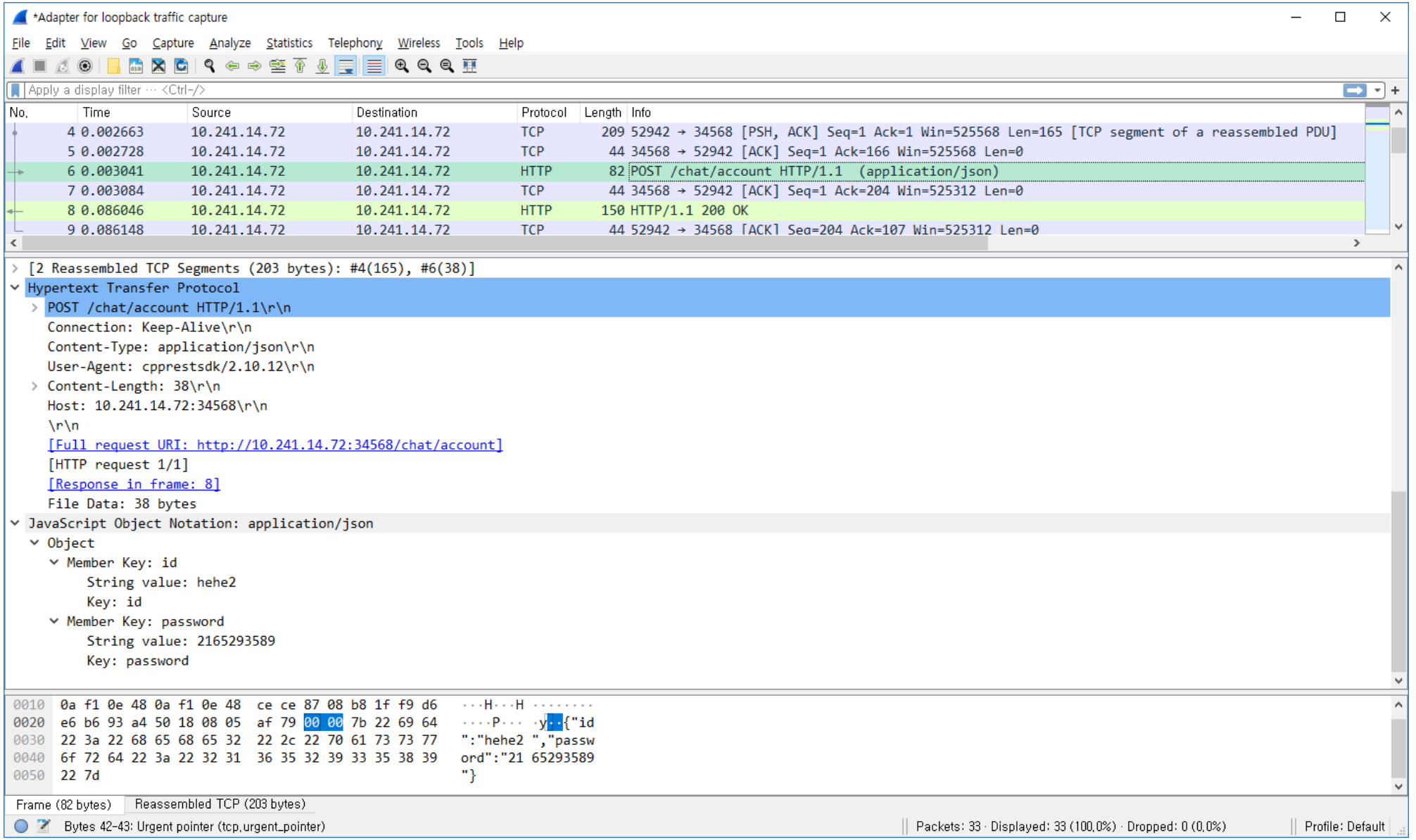
- User signup Test



팀프로젝트 | RestAPI verify tool(wireshark)



팀프로젝트 | RestAPI verify tool(wireshark)



Chat server/client Guide

Template repository 참고

Best reviewer 양성과정

Group Mentoring

2일차 진행됨

Group Mentoring(팀프로젝트 기간 중 1일:화요일)

- 멘토링은 핵심 가치나 조직문화를 강화/유지하는데 기여하고, 공통의 문화적 가치나 회사가 기대하는 바를 구성원들의 마음 속에 심어 줌

- 멘토링 그라운드 룰
  - 질문을 하자(There is no stupid question)
  - 질문을 하자
  - 질문을 하자
  - 멘토의 의견을 듣고, 결정은 우리가 하도록 하자
  - 함께 생각하고 고민하는 멤버이며, 먼저 경험을 해본 동료로써 받아들이자
  - 직접 coding이나 commit을 원하지는 말자
  - 멘토의 의견에 환경과 절차, 내용에 제약을 두지 말자
    - 직접적인 1차 코드 리뷰, comment에 대한 comment, merge 완료 된 것에 대한 comment, code inspection을 통한 사후 리뷰
    - 사소한 이슈에 대한 리뷰나 의견
  - 프로젝트 요구사항의 완성보다는 코드 품질 대해 더 관심과 의견을 얻자
  - 코드 품질에 관한 debate를 중요하게 생각하자

스케쥴

- 멘토 1인 2개의 팀을 멘토링하며, 4~5명 1팀으로 구성되어 있음
- 상세 시간 운영 계획

시 간	내 용
09:00 ~ 10:00	1팀 팀프로젝트 녹스미팅
10:00 ~ 11:00	2팀 팀프로젝트 녹스미팅
11:00 ~ 16:30	개별 멘토링 - 개별 질문에 답변(There is no stupid question) - Open PR에 대한 리뷰(설계, 가독성, Testability 등 코드 품질 관점) - Close PR 에 대한 리뷰 - 전체 코드에 대한 Inspection
16:30 ~ 17:30	멘토끼리 하는 회고(운영자, 멘토만 참여)

Q & A

Team Project : 회고

4일차 오후에 진행됨

회고란?

- Retrospective, Reflection, Postmortem 등

- 팀이 정해진 기간 동안 해왔던 일에 대해 돌아보며,
  팀원들이 함께 생각하고 배우는 시간
  “ 팀 단위 자기 성찰 "

- 팀은 회고를 하면서 눈 앞에 보이는 이익뿐 아니라
  능력과 기쁨이 증가하는 경험을 맛볼 수 있다
  – 애자일 회고

회고를 어떻게 하나요?

(회고 진행자 정하기)

- 각자 현재 감정/상태와 간략한 이유 말하기 (3분)
  - 서로 각자의 상태를 파악하고, 발언 시 감정적으로 되지 않도록 주의
- 자료 모으기 (5분)
- KPT로 정리 (25분)
  - Keep, Problem, Try

좋은 회고를 위한 Tip.

- 회고 시 기술적 측면과 사람 측면을 함께 다룬다

  “프로젝트” vs “팀워크”

- 심리적인 안전감이 중요

  솔직하게 이야기하되 가혹해지면 안 된다

  (더 작고, 더 목표지향적이고, 덜 개인적이고, 덜 비판적이면서 영향력 있는 피드백)

- 회고에서 나온 것을 실행하려는 의지

  아무리 좋은 의견이 나와도 이를 실천하지 않으면 ‘꽝’ 이다

Check-In (3분)

- 돌아가면서 현재 자신의 컨디션을 1~5로 표현하고 간략한 이유를 말해봅시다

- Ex)

  “ 4 , 오랜만에 교육을 들으니 설레요,

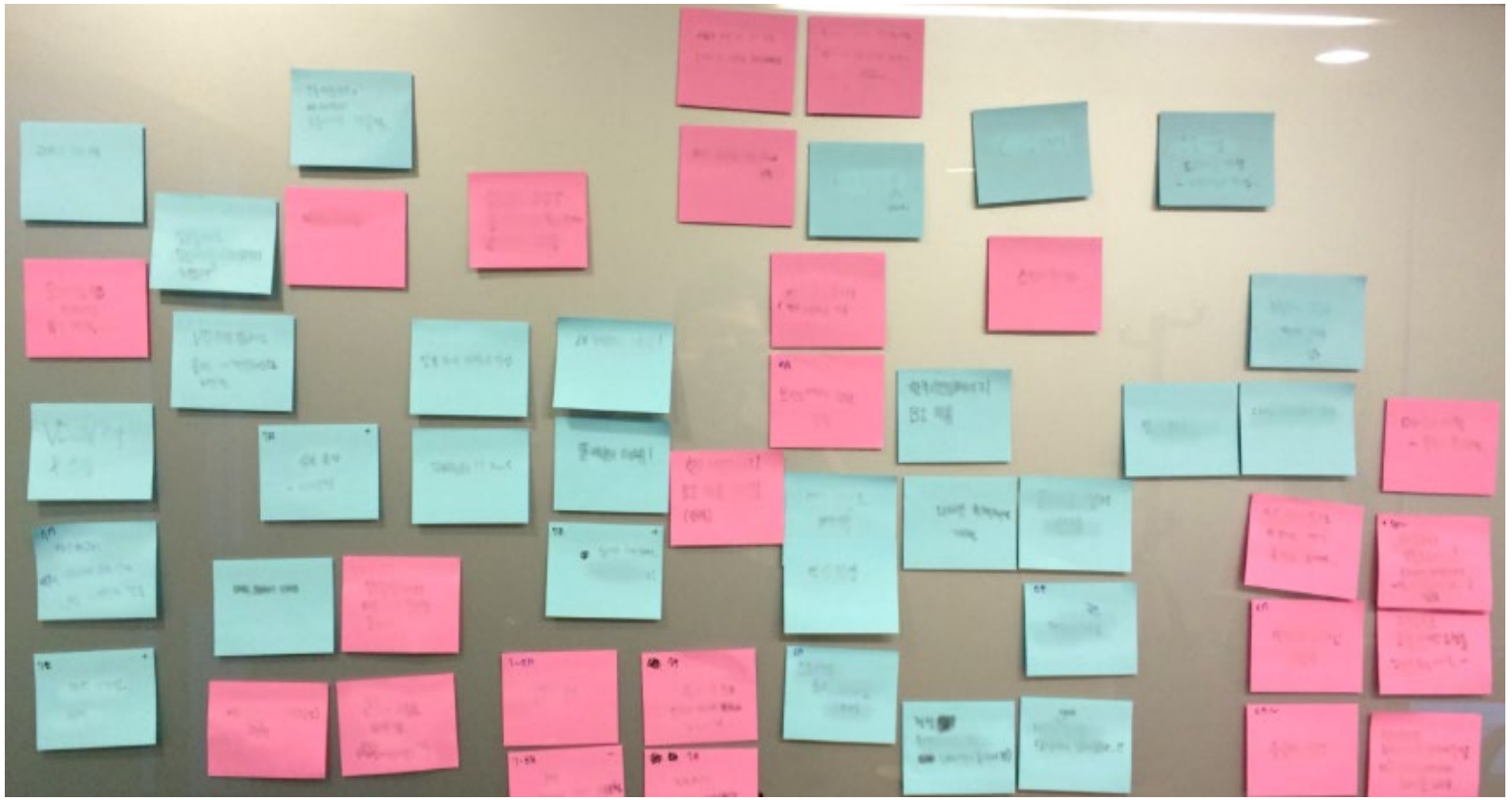
  5점이 아닌 이유는 배가 고파서요 “

  “ 2 , 늦게까지 프로젝트 하느라 잠을 못 잤더니 피곤해서요,

  1점이 아닌 이유는 퇴근할 시간이 다되어서 좀 기뻐요 “

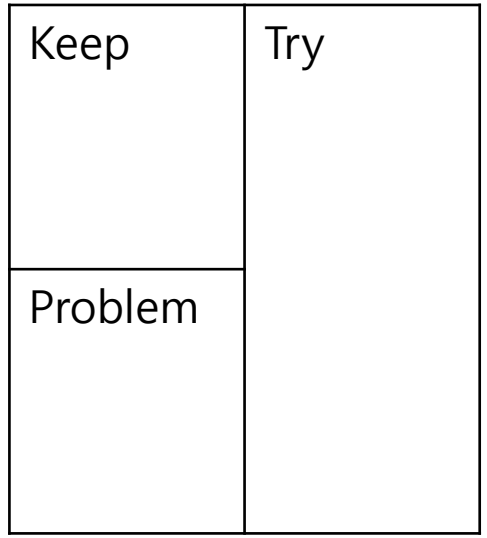
자료 모으기 (5분)

- “좋았던 점 " 과 “아쉬웠던 점 " 을 작성합니다
- 한 포스트잇에는 한 가지 내용만 적습니다
- 인당 3개 이상 작성합니다



KPT로 정리 (25분)

- 이젤 패드를 Keep, Problem, Try 세 부분으로 나눕니다



- Keep (5분)
  - 좋았던 점을 기반으로 도출한 앞으로 계속 유지해야할 사항
- Problem (5분)
  - 아쉬웠던 점을 기반으로 도출한 앞으로 개선되어야 할 사항
- Try (15분)
  - 도출된 Problem의 원인을 파악하여 이를 기반으로 어떤 시도들을 해볼 수 있는지
  - 구체적인 Action item 선정

팀 프로젝트 발표 안내(발표는 5일차 오후 진행)

- 발표 자료를 확인할 수 있는 방법을 팀별 repository root (master branch)에 올려주시기 바랍니다
- 발표 순서
  - 랜덤으로 선정
- 발표 내용 (팀별 15분 이내, 발표자는 N명 가능)
  - 진행/구현 방법(1분)
    - 코딩 그라운드 룰, 코딩 스타일, 리뷰 정책, branch 정책,
    - 구조 설계 변경, 요구사항 분배, 구현 시 온라인 협업 등
  - 주요 구현/리뷰 사항(10분)
    - PR 단위
  - 회고 및 느낀점 (3분)