Backend 🡨 🡪 frontend

* Backend🡪 url

@GetMapping("/todos")

public List<Todo> getAllTodos() {

* Json

@PostMapping("/todos")

public Todo createTodo(@Valid @RequestBody Todo todo) {

@GetMapping(value="/todos/{id}")

public ResponseEntity<Todo> getTodoById(@PathVariable("id") String id) {

@PutMapping(value="/todos/{id}")

public ResponseEntity<Todo> updateTodo

@DeleteMapping(value="/todos/{id}")

public ResponseEntity<?> deleteTodo(@PathVariable("id") String id) {

* Frontend : lấy dữ liệu+hành động người dung🡪 url tương ứng với method=get/post(create)/put(update)/delete 🡪 backend nhận tương ứng từng request với url +parameter🡪…

🡪coi rest api căn bản.//

Angular: Service.ts

@Injectable()

export class TodoService {

private baseUrl = '[http://localhost:8080](http://localhost:8080/)';

constructor(private http: Http) { }

getTodos(): Promise<Todo[]> {

return this.http.get(this.baseUrl + '/api/todos/')

.toPromise()

.then(response => response.json() as Todo[])

.catch(this.handleError);

}

createTodo(todoData: Todo): Promise<Todo> {

return this.http.post(this.baseUrl + '/api/todos/', todoData)

.toPromise().then(response => response.json() as Todo)

.catch(this.handleError);

}

updateTodo(todoData: Todo): Promise<Todo> {

return this.http.put(this.baseUrl + '/api/todos/' + todoData.id, todoData)

.toPromise()

.then(response => response.json() as Todo)

.catch(this.handleError);

}

deleteTodo(id: string): Promise<any> {

return this.http.delete(this.baseUrl + '/api/todos/' + id)

.toPromise()

.catch(this.handleError);

}

private handleError(error: any): Promise<any> {

console.error('Some error occured', error);

return Promise.reject(error.message || error);

}

}

#### Túm lại: thay vì dùng method http bình thường: get/post lấy dữ liệu từ from . Ta dùng rest api

+ tách riêng biệt front end với backend (frontend: ko cần sửa code bên trong khi chuyển qua backend, + 2 bên làm song song+ frontend: url giả chứa method ok)