

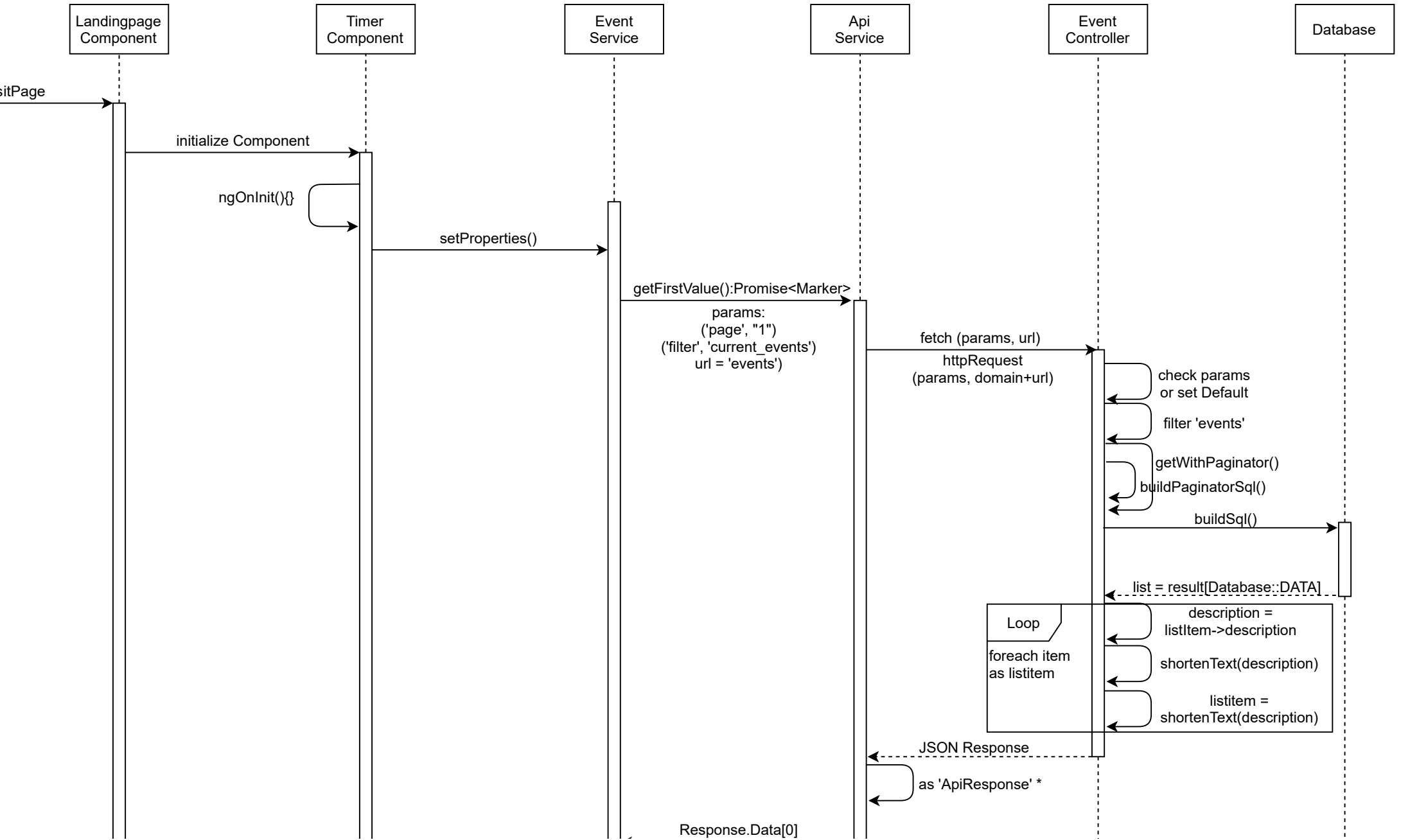
User

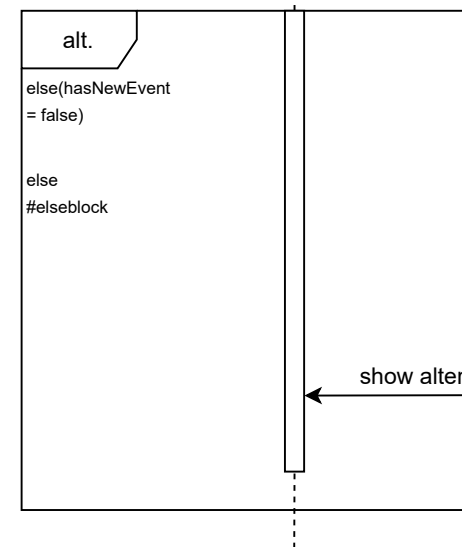
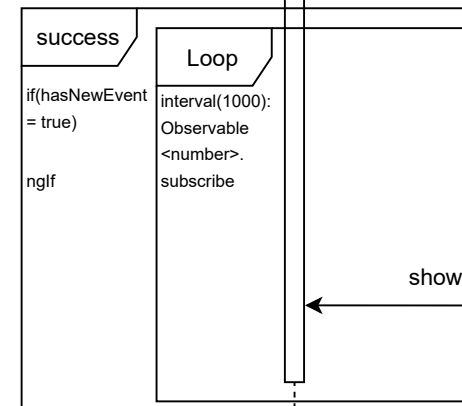


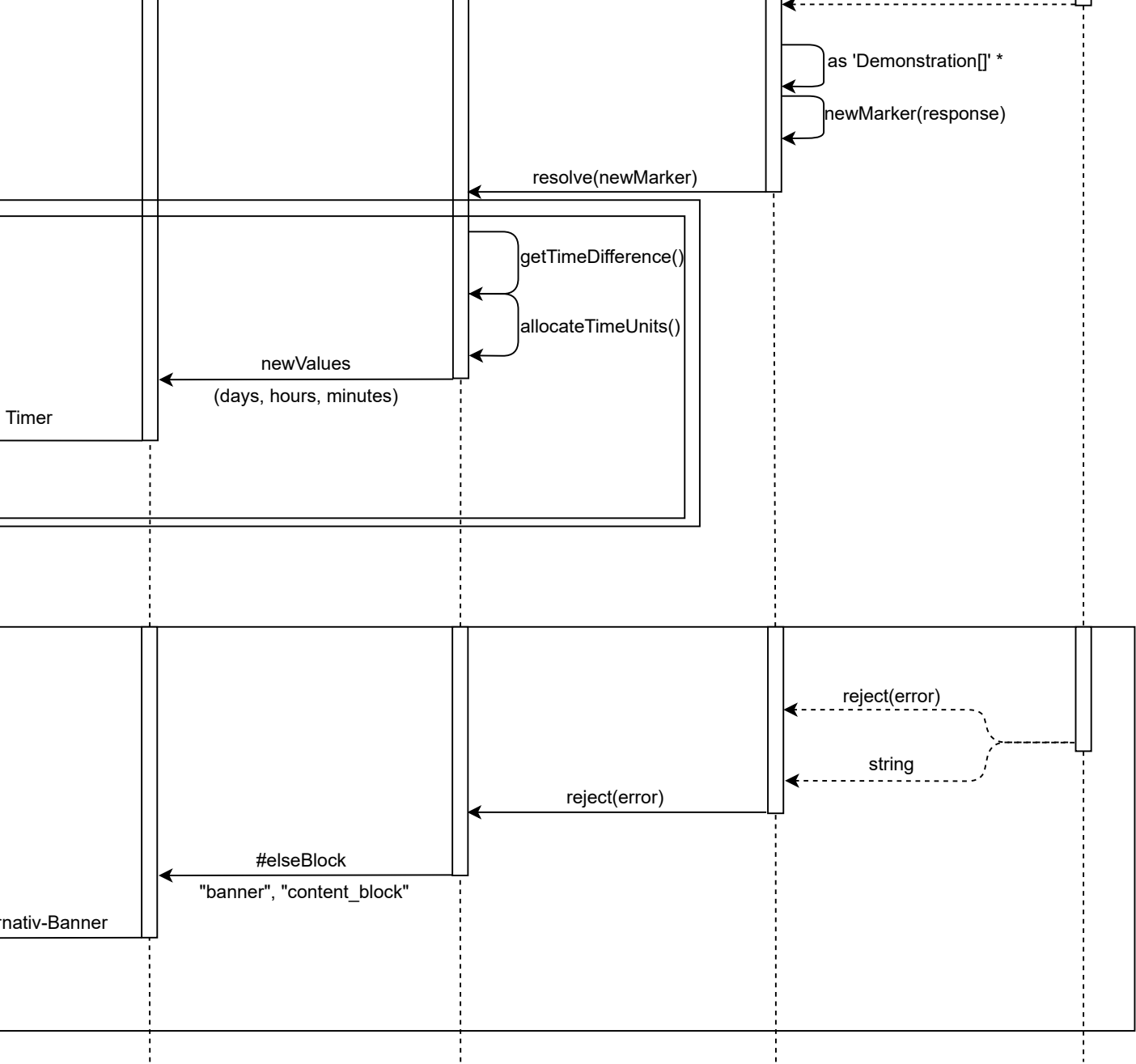
vis

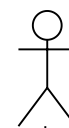


Timer





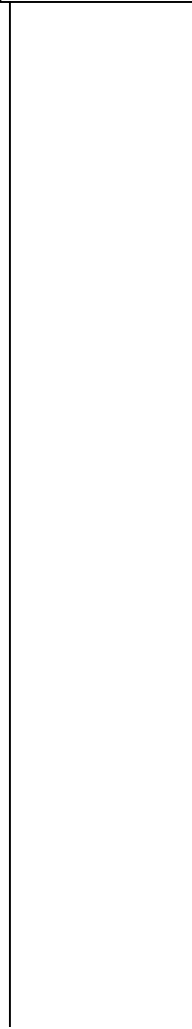




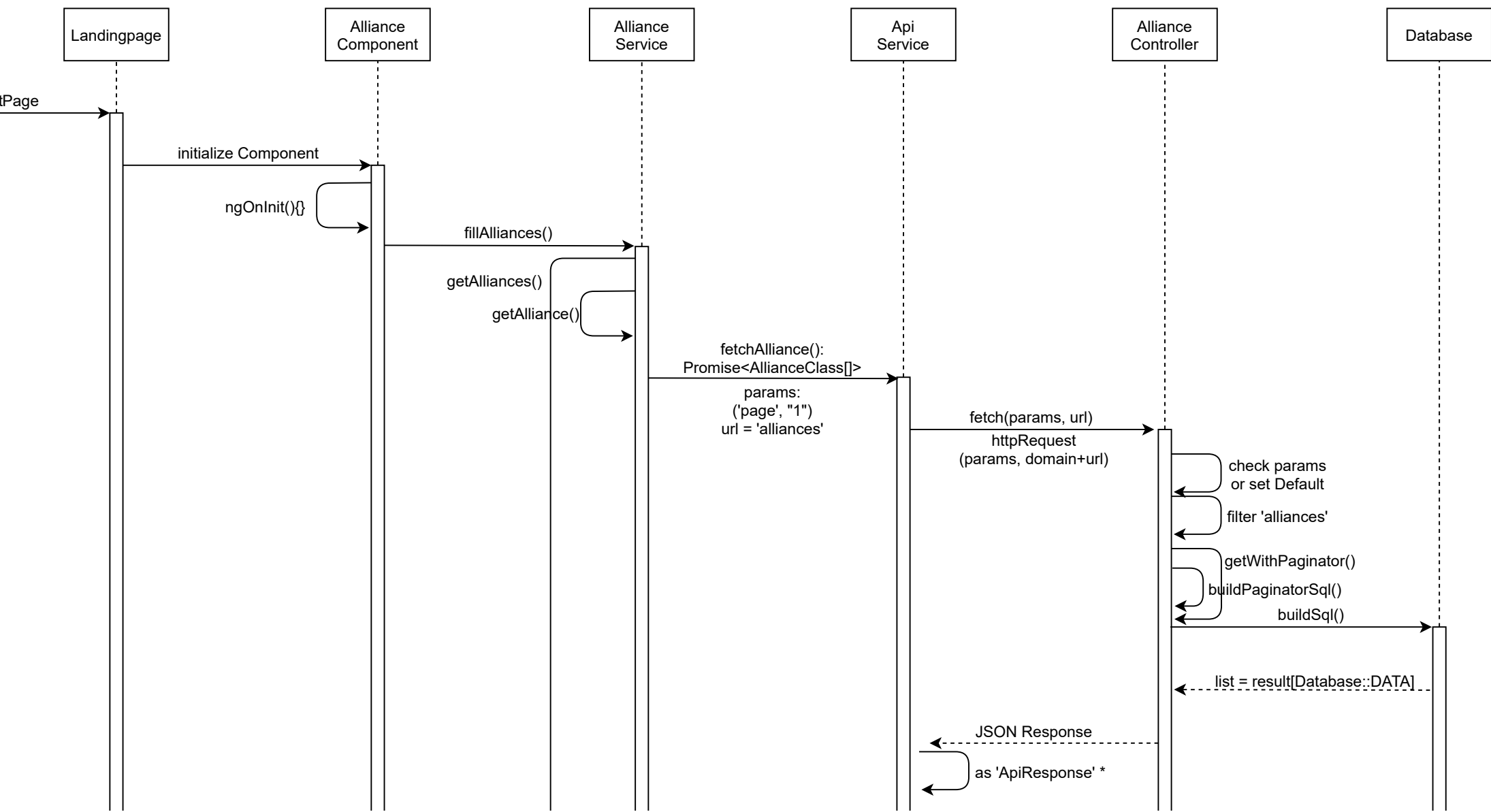
User

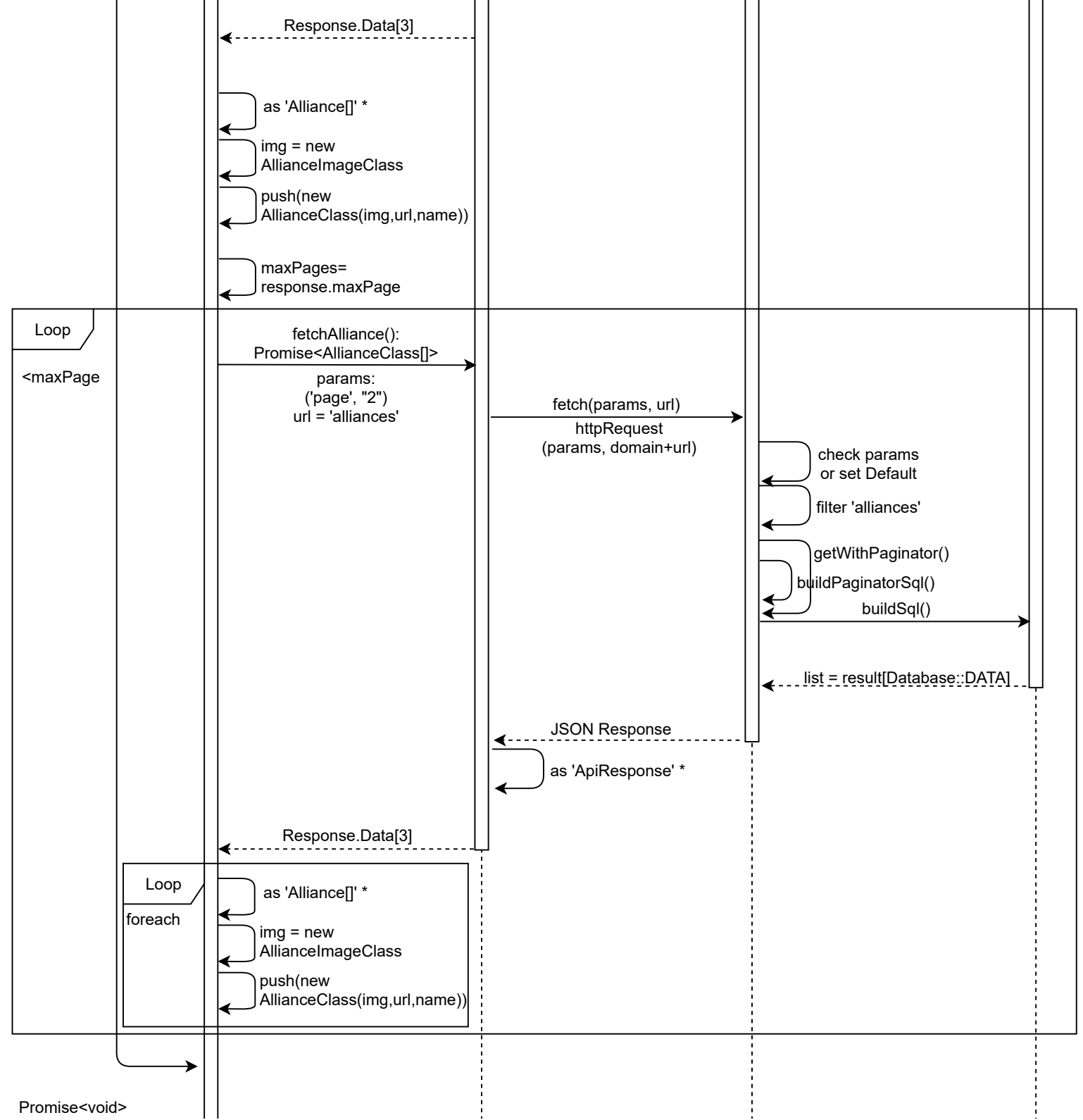


visit



Alliance

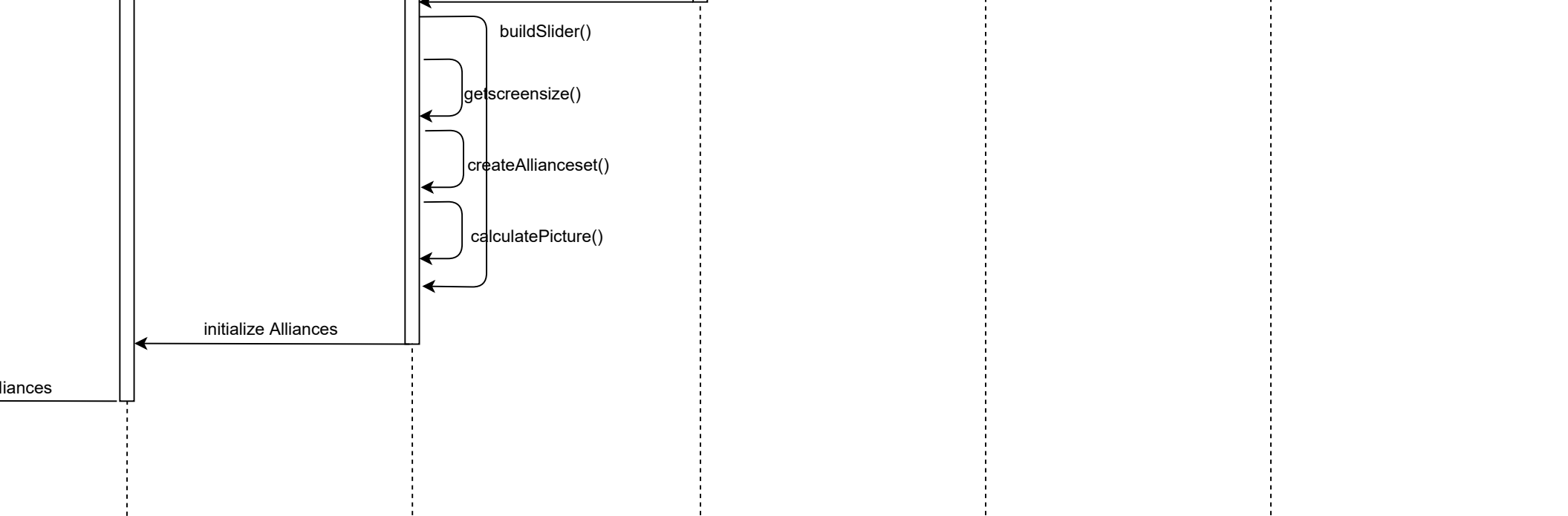




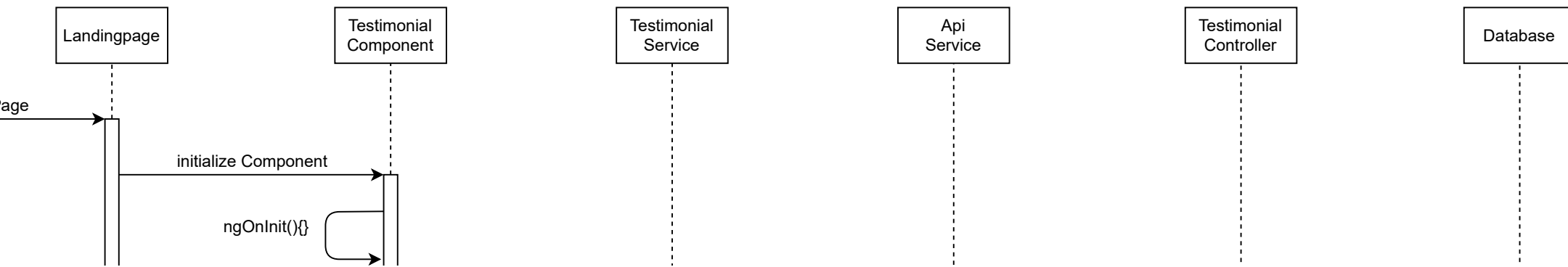
show All

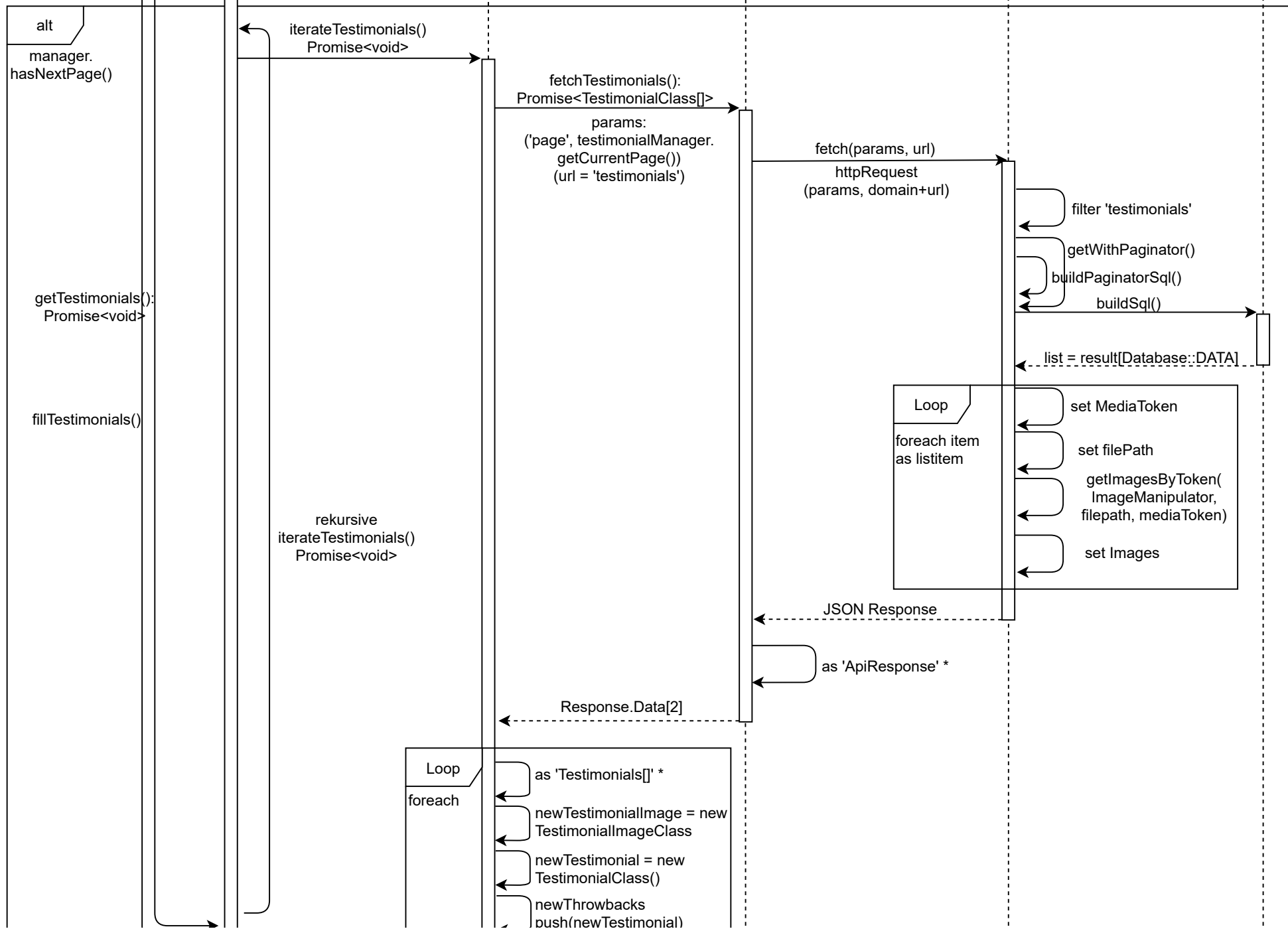


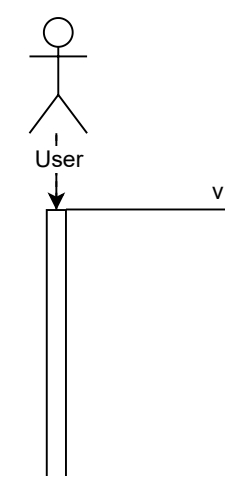
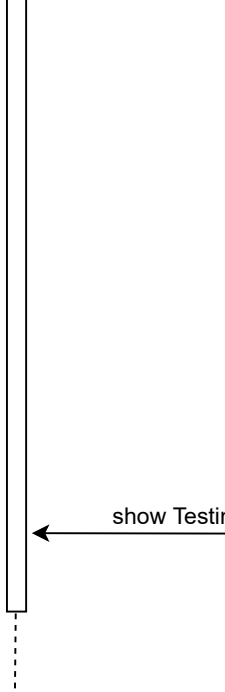
visitP

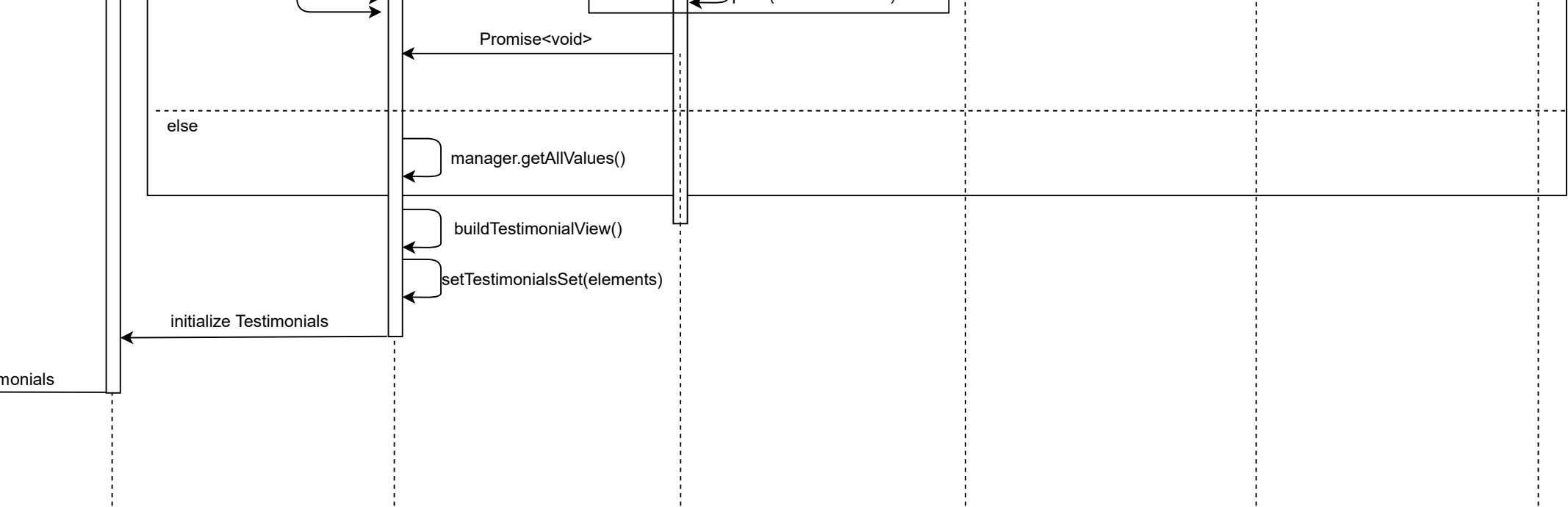


Testimonials

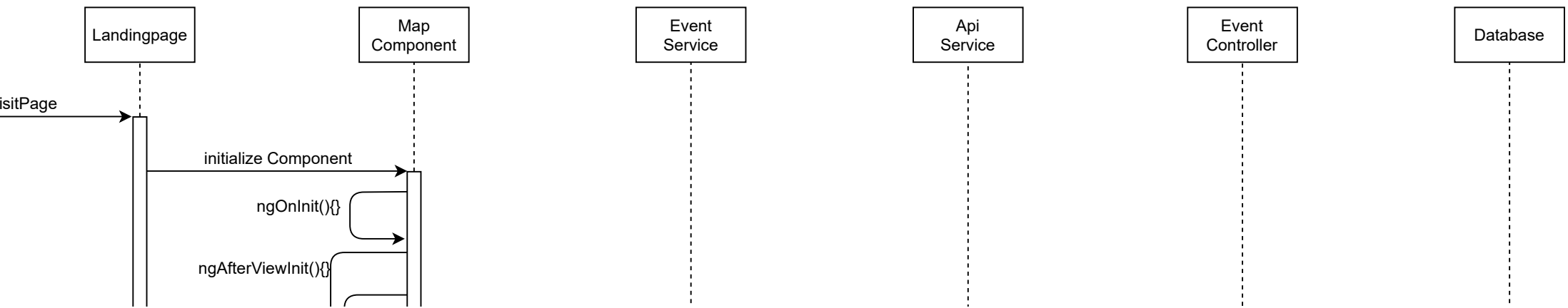


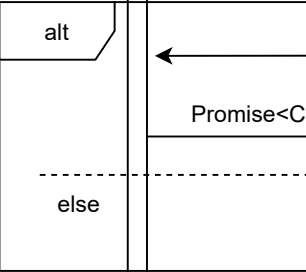


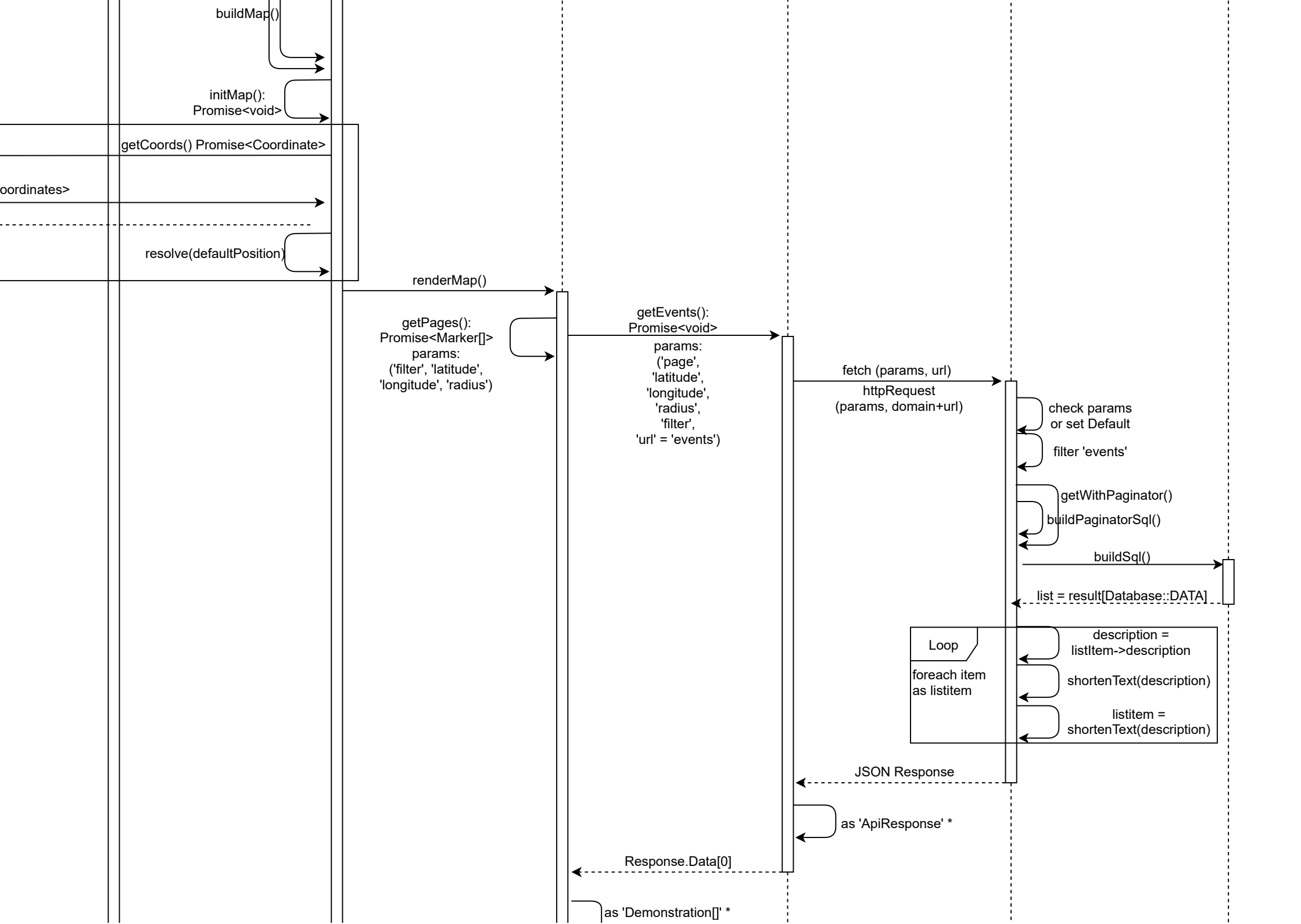


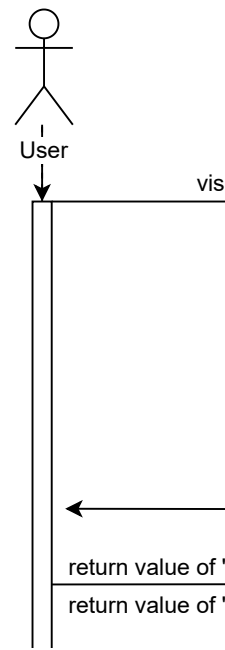
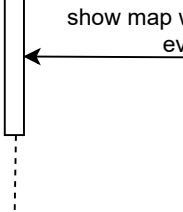


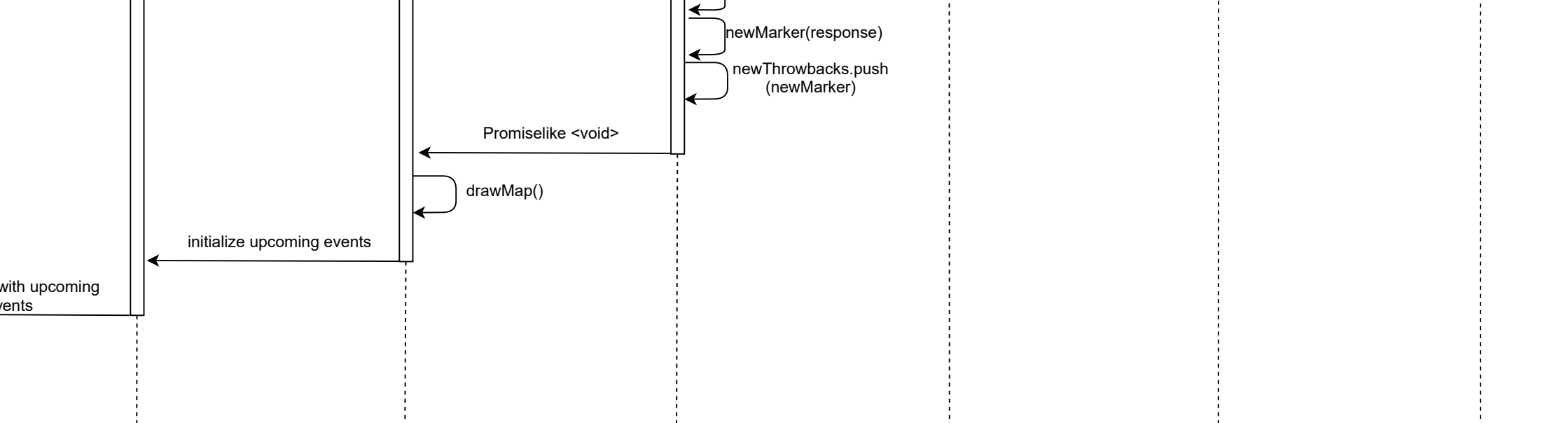
Map



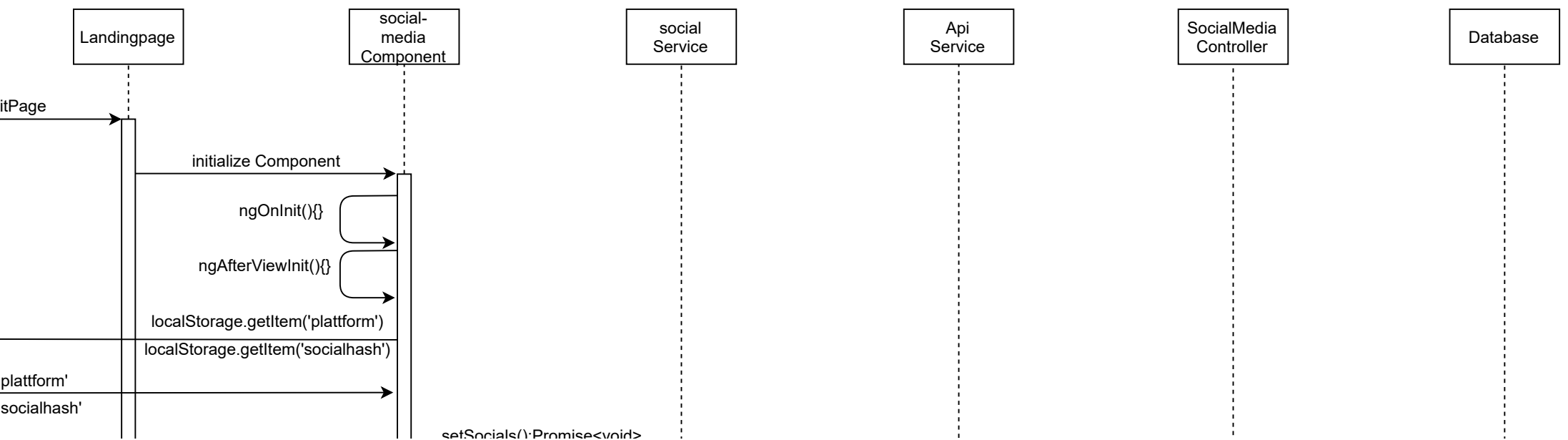


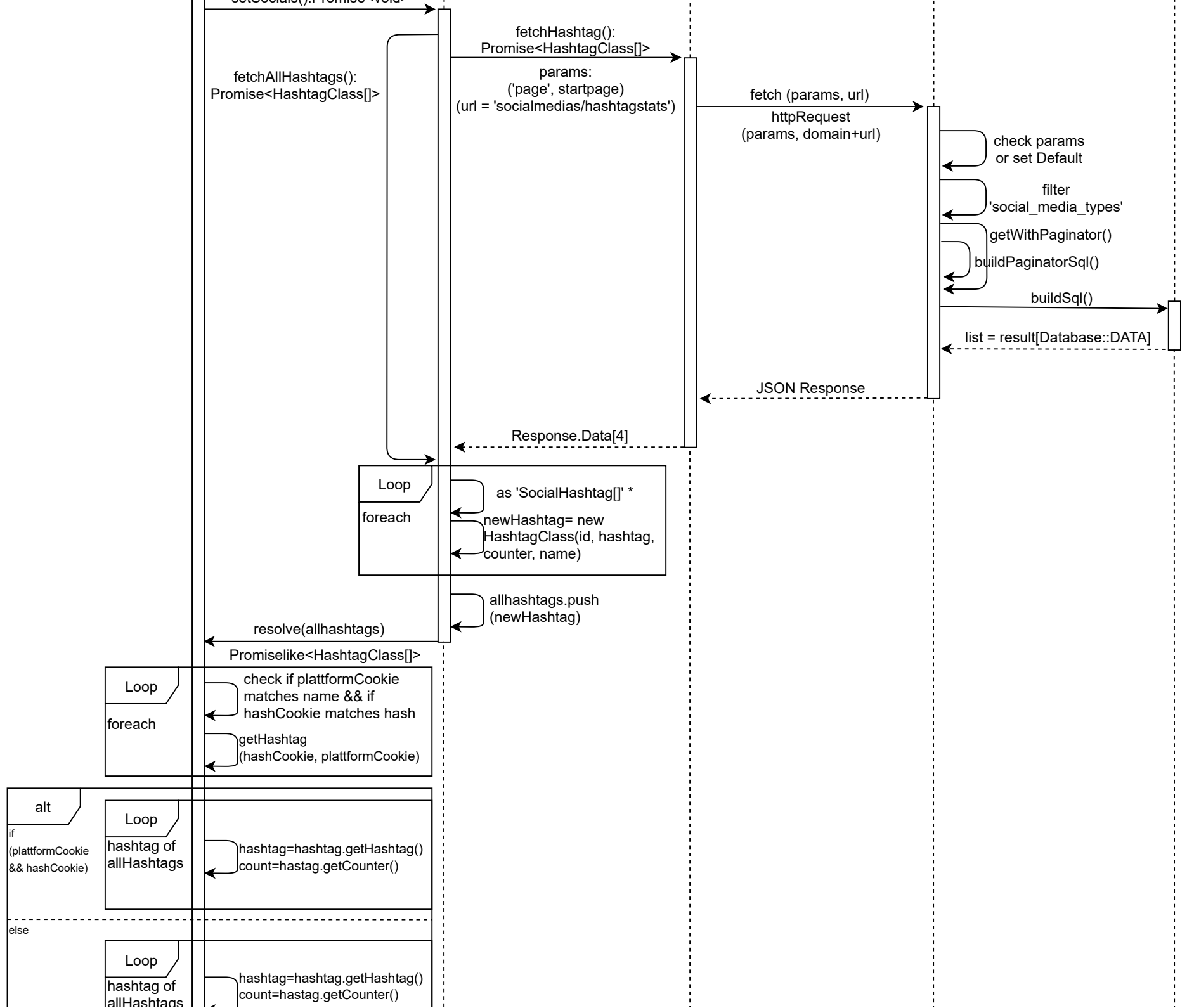




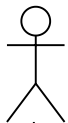
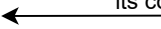


Social Media





show the specific
its co

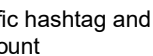


User



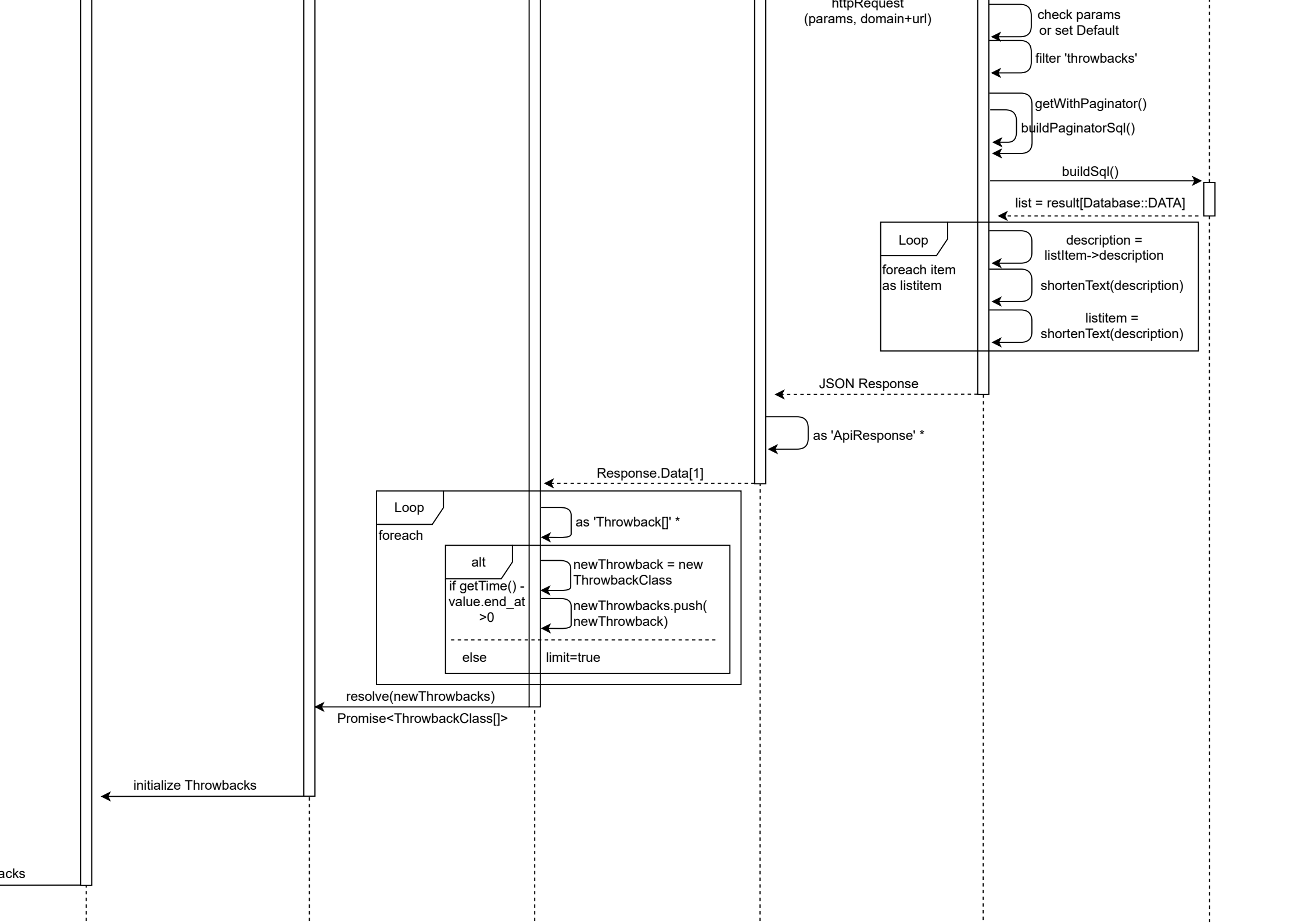
visitPe





Throwback





1

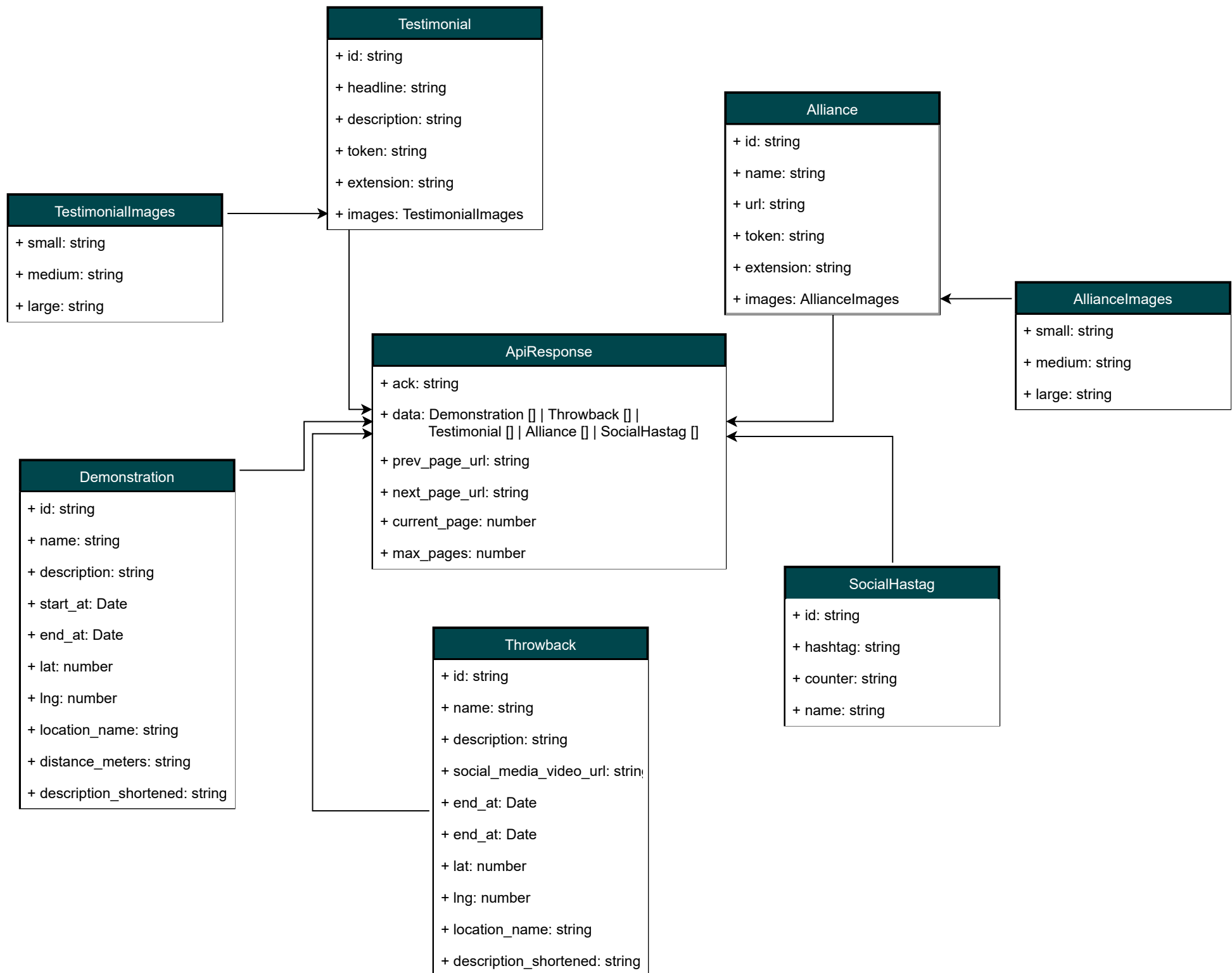
2

3

4

5

6



active.component
constructor(private viewport: viewportService)

alliance.component
- scrHeight: number - scrWidth: number + urls: SafeUrl [] + manyEntress: number + alliances: Map<number, AllianceClass []> + hasAlliance: boolean
+ getScreenSize() + async fillAlliances() + buildSlider() + buildSlider() + calculatePicture() + createAllianceSet(srcWidth: number) constructor(private allianceService: AllianceService, private viewport: ViewportService) constructor(private allianceService: AllianceService, private viewport: ViewportService)

faq.component
constructor(private viewport: ViewportService)

fillerone.component
constructor(private viewport: ViewportService)

fillertwo.component
constructor(private viewport: ViewportService)

fillerthree.component
constructor(private viewport: ViewportService)

fillerfour.component
constructor(private viewport: ViewportService)

fillerfive.component
constructor(private viewport: ViewportService)

fillersix.component
constructor(private viewport: ViewportService)

footer.component
constructor(private modalService: NgbModal, private cookieService: CookieService) + open(name: string) - getDismissReason(reason: any): string

landingpage.component
constructor()

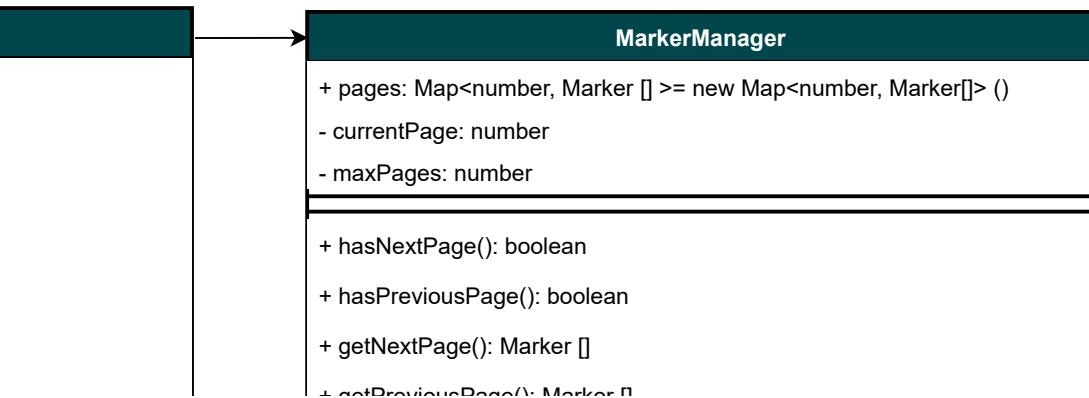
map.component
<div><div><div>- userPosition: Coordinate</div><div>- mapView: View</div><div>- map: any</div><div>- markers: Marker []</div><div>+ place</div><div>+ day</div><div>+ error</div><div>+ searchedLocationValue: string</div><div>+ mapSuccess: boolean</div><div>+ mapSliderValue: number</div><div>+ sliderChanged: boolean</div><div>- cluster: VectorLayer</div><div>- overlay: Overlay</div></div></div>
<div><div><div>+ async buildMap()</div><div>- initMap(zoom: number): Promise<void></div><div>+ async getCoords(): Promise<Coordinate></div><div>+ async renderMap(radius: number)</div><div>- markerCluster(marker: Marker [])</div><div>- drawMap(marker: Marker [], distance: number, zoom: number, mapScreen: Map)</div><div>+searchLocation()</div><div>+ deleteValue()</div><div>+ getSliderValue(value: number)</div><div>constructor(private eventService: EventService, private viewport: ViewportService)</div></div></div>

navigation.component
<div><div>constructor()</div></div>

social-media.component
<div><div><div>+ count: string</div><div>+ hashtagInstagram: string</div><div>+ hashtagFacebook: string</div><div>+ hashtag: string</div><div>+ hashtagSuccess: boolean</div><div>+ allHashtags: HashtagClass []</div><div>+ model: string</div><div>- platformCookie: string null</div><div>- hashCookie: string null</div></div></div>
<div><div><div>+ async setSocials()</div><div>+ getHashtag(name: string, platform: string)</div><div>+ deleteChecked()</div><div>constructor(private viewport: ViewportService, private socialService: SocialService, private cookieService: CookieService)</div></div></div>

throwbacks.component
<div><div><div>+ closeResult: string</div><div>+ modalOptions: NgbModalOptions</div><div>+ throwbackPages: ThrowbackClass []</div><div>+ maxPage: number</div><div>+ page: number</div><div>+ hasThrowback: boolean</div><div>+ videoSuccess: boolean</div></div></div>
<div><div><div>+ open(content: any)</div><div>- getDismissReason(reason: any): string</div><div>+ async setProperties(page?: number)</div><div>+ setMaxPage(size: number)</div><div>constructor(private throwbackService: ThrowbackService, private viewport: ViewportService, private modalService: NgbModal)</div></div></div>

Marker
<div><div><div>- id: string</div><div>- name: string</div><div>- description: string</div><div>- startDate: Date</div><div>- endDate: Date</div><div>- lat: number</div><div>- lng: number</div><div>- locationName: string</div></div></div>

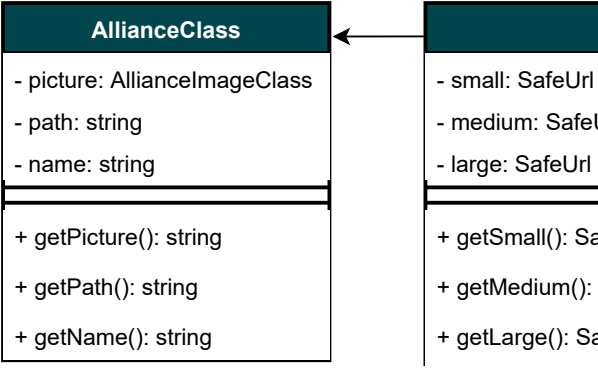


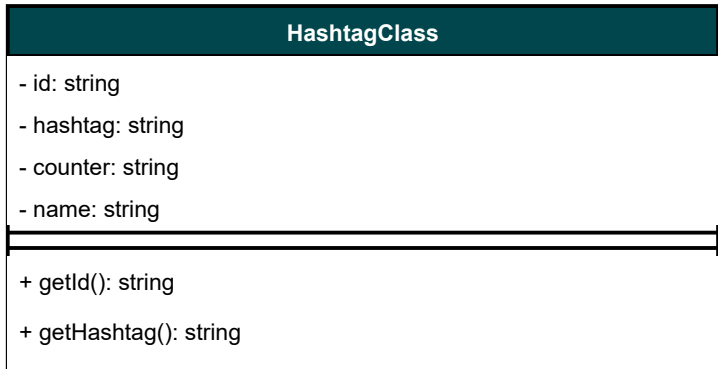
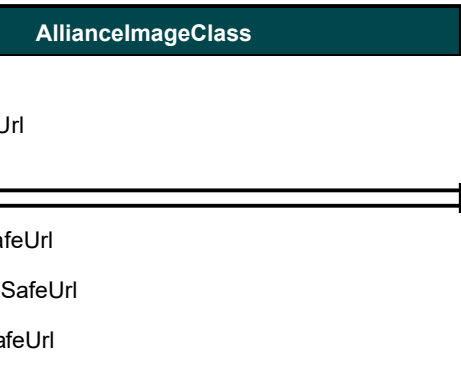
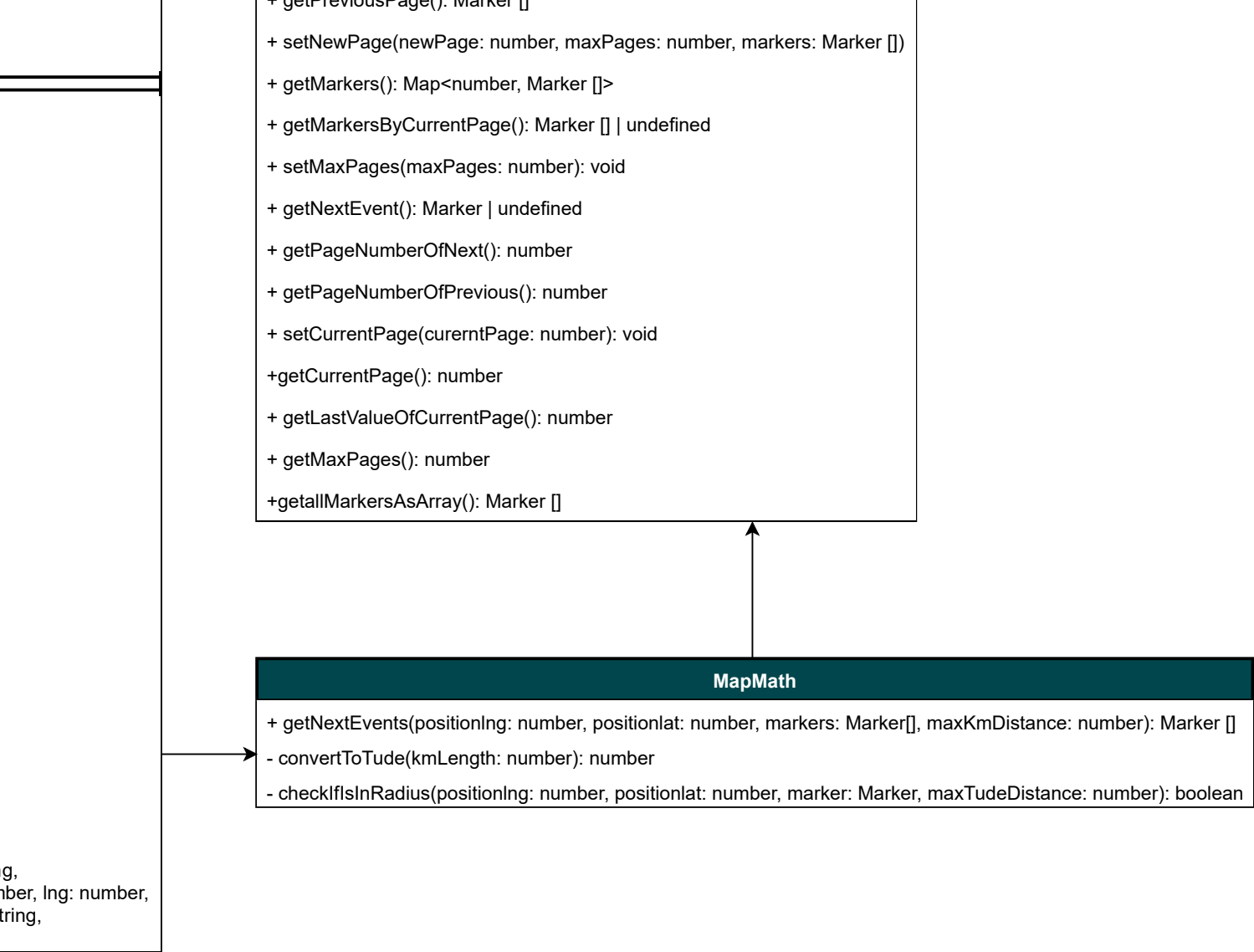
testimonials.component
+ scrHeight: number + scrWidth: number + manyEntress: number - allTestimonials: TestimonialClass [] + chosenTestimonials: TestimonialClass [] + urls: SafeUrl [] + hasTestimonials: boolean + hasMore: boolean + setOfTestimonials: Map<number, TestimonialClass []>
+ async setSocials() + getHashtag(name: string, plattform: string) + deleteChecked() + async fillTestimonials() + async buildTestimonialView() + async getTestimonials() + async iterateTestimonials + setTestimonialsSet(elements: number) + getScreenSize() constructor(private testimonialService: TestimonialService, private viewport: ViewportService, public domsnitizer: DomSanitizer)

modal-focus
+ open(name: string) - getDismissReason(reason: any): string constructor(private modalService: NgbModal, private cookieService: CookieService)

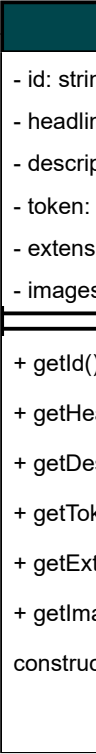
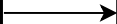
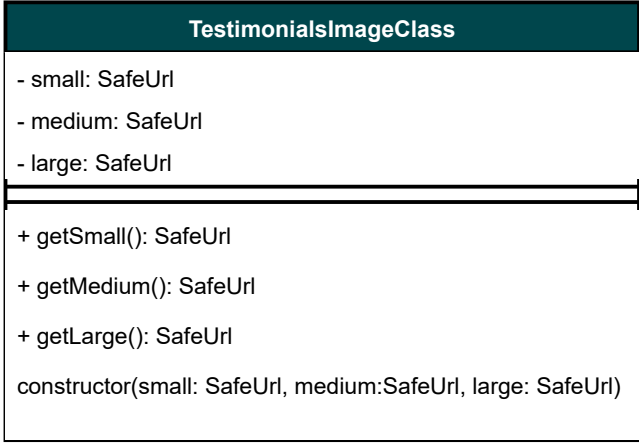
timer.component
- subscription: Subscription + dateNow: Date + dDay: Date + timeDifference: number + secondsToDay: number + minutesToDay: number + hoursToDay: number + daysToDay: number + gDate: string + location: string + hasNewEvent: boolean - milliSecondsInASecond: number - hoursInADay: number - minutesInAnHour: number - SecondsInAMinute: number
+ async setProperties - getTimeDifference(newDate?: Date) - allocateTimeUnits(timeDifference: number) constructor(private eventService: EventService, private viewport: ViewportService)

- descriptionShortened: string
- distanceMeters: string
+ getId(): string
+ setId(id: string): void
+ getName(): string
+ setName(name: string): void
+ getDescription(): string
+ setDescription(description: string): void
+ getStartDate(): string
+ setStartDate(startDate: Date): void
+ getEndDate(): string
+ setEndDate(endDate: Date): void
+ getLat(): number
+ setLat(lat: number): void
+ getLng(): number
+ setLng(lng: number): void
+ getLocationName(): string
+ setLocationName(locationName: string): void
+ getDescriptionShortened(): string
+ getDistanceMeters(): string
constructor(id: string, name: string, description: string, startDate: Date, endDate: Date, lat: number, lng: number, locationName: string, distanceMeters: string, descriptionShortened: string)





constructor(sma



all: SafeUrl, medium: SafeUrl, large: SafeUrl)

+ getCounter(): string
+ getName(): string
constructor(id: string, hashtag: string, counter: string, name: string)

TestimonialClass

ing
ne: string
tion: string
string
ion: string
s: TestimonialsImageClass

: string
adline(): string
scription(): string
ken(): string
tension(): string
ages(): TestimonialImageClass
ctor(id: string, headline: string, description: string,
token: string, extension: string,
images: TestimonialImageClass)

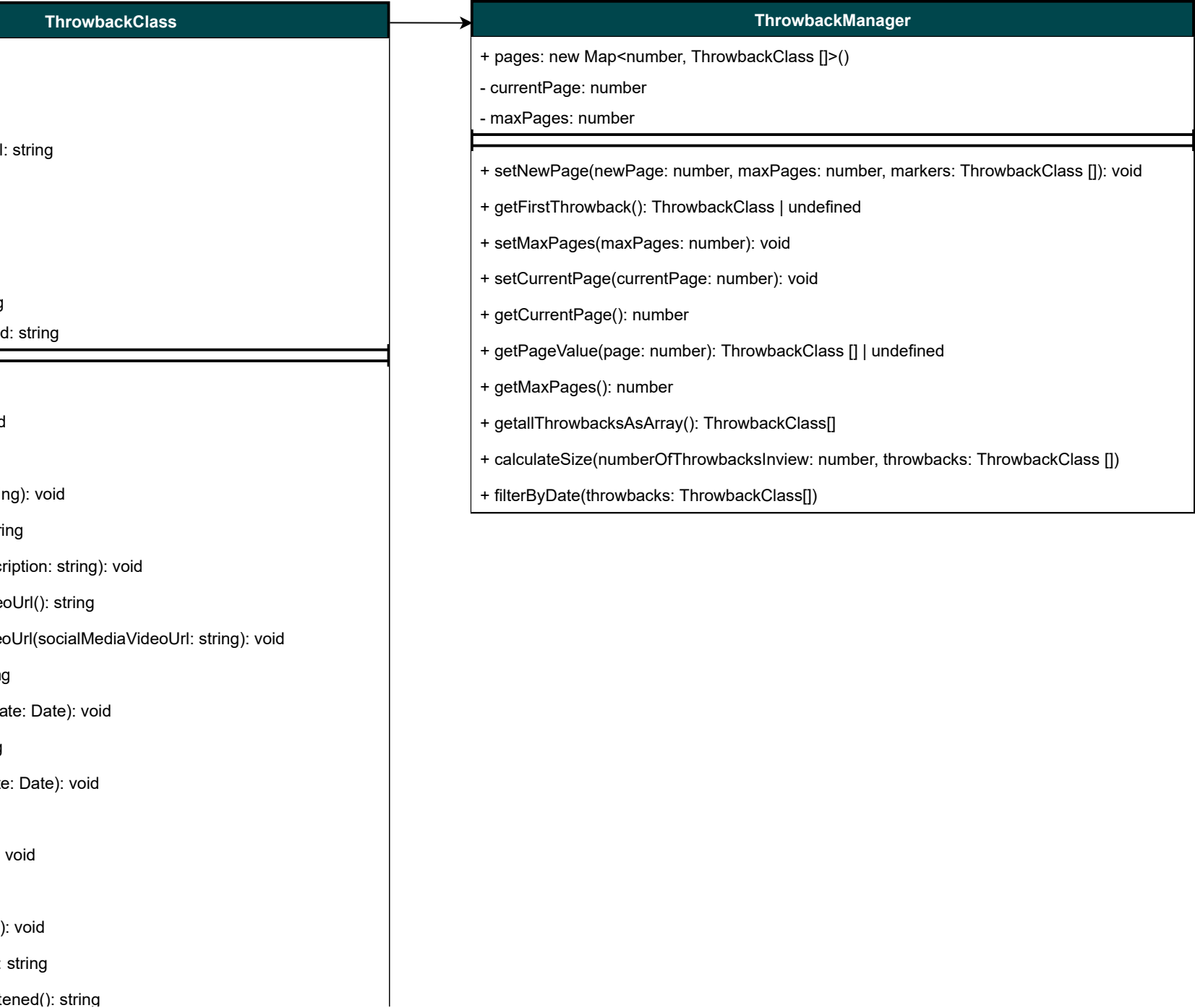
TestimonialManager

+ pages: Map<number, TestimonialClass [] >
- currentPage: number
- maxPages: number

+ hasNextPage(): boolean
+ hasPreviousPage(): boolean
+ getNextPageValue(): TestimonialClass []
+ getPreviousPageValue(): TestimonialClass [9]
+ setNewPage(newPage: number, maxPages: number, Testimonials: TestimonialClass []): void
+ getTestimonials(): Map<number, TestimonialClass []>
+ getTestimonialsByCurrentPage(): TestimonialClass [] | undefined
+ setMaxPages(maxpages: number): void
+ getNextEvent(): TestimonialClass | undefined
+ getManyTestimonialsAsPages(elementsPerPage: number): Map<number, TestimonialClass []>
+ getPageValue(page: number): TestimonialClass[] | undefined
+ getPageNumberOfNext(): number
+ getPageNumberOfPrevious(): number
+ setCurrentPage(currentPage: number): void
+ getCurrentPage(): number
+ getLastValueOfCurrentPage(): number
+ getMaxPages(): number
+ getAllValues(): TestimonialClass[]
+ getNextPage(): number
+ getPreviousPage(): number
constructor()

- id: string
- name: string
- description: string
- socialMediaVideoUr
- startDate: Date
- endDate: Date
- lat: number
- lng: number
- locationName: string
- descriptionShortene

+ getId(): string
+ setId(id: string): void
+ getName(): string
+ setName(name: string): void
+ getDescription(): string
+ setDescription(description: string): void
+ getSocialMediaVideoUr
+ setSocialMediaVideoUr
+ getStartDate(): string
+ setStartDate(startDate: Date): void
+ getendDate(): string
+ setendDate(endDate: Date): void
+ getLat(): number
+ setLat(lat: number): void
+ getLng(): number
+ setLng(lng: number): void
+ getLocationName(): string
+ getDescriptionShort



+ setDescriptionShort
constructor(id: string,
socialMed
lat: numb
description

Shortened(short: string): void

name: string, description: string,
mediaVideoUrl: string, startDate: Date, endDate: Date,
er, lng: number, locationName: string,
onShortened: string)

