

AnnotationEmitterService

September 4, 2026

AnnotationEmitterService

Kind: Service

Source: `atloria-monorepo/apps/screenshot-worker/src/app/screenshot/services/annotation-emitter.service.ts`

TS port of the atlorix Python annotation producer
(tools/capture_annotations.py + tools/emit_annotations.py).

Locates a step's target element on the LIVE page, converts it to percent-of-natural-image coordinates, and builds the non-destructive overlay payload + the `#atloria-annotations=` URL fragment the web viewer consumes. Coordinates only — the screenshot pixels are never touched (moat intact).

`AnnotationEmitterService` locates a screenshot step's target element on the live page and converts its bounding box into percentage coordinates relative to the natural screenshot image. It produces non-destructive annotation overlay data and the `#atloria-annotations=` URL fragment consumed by the web viewer, without modifying screenshot pixels.

Methods

Method	Signature	Returns	Description
<code>recomputeTargetBoxPct</code>	<code>recomputeTargetBoxPct(page: Page, selector: string, opts: { fullPage: boolean })</code>	<code>Promise<BoxPct></code>	<code>null></code>
<code>buildStepAnnotations</code>	<code>buildStepAnnotations(boxPct: BoxPct, stepIndex: number)</code>	<code>Annotation[]</code>	Turn a located box into a <code>highlight box</code> + a numbered <code>marker</code> at the box center.

Method	Signature	Returns	Description
toFragment	toFragment(annotations: Annotation[])	string	Build the #atloria-annotations=<url-encoded JSON> fragment for a set of annotations.

Where it refuses work

- AnnotationEmitterService stops the work with an early return when !box .
- AnnotationEmitterService stops the work with an early return when box.w < 0.5 || box.h < 0.5 .
- AnnotationEmitterService stops the work with an early return when box.x < 0 || box.y < 0 || box.x > 100 || box.y > 100 || box.x + box.w > 100 || box.y + bo... .
- AnnotationEmitterService stops the work with an early return when !annotations || annotations.length === 0 .

When something fails

- AnnotationEmitterService handles failure in 1 place: it turns it into a return value in all 1.

Diagram

```
sequenceDiagram
    participant Caller as Screenshot Workflow
    participant Service as AnnotationEmitterService
    participant Page as Live Browser Page
    participant Viewer as Web Viewer

    Caller->>Service: recomputeTargetBoxPct()
    Service->>Page: Locate target element
    Page-->>Service: Element bounding box
    Service->>Page: Read natural image/page dimensions
    Page-->>Service: Dimensions
    Service-->>Caller: BoxPct | null

    Caller->>Service: buildStepAnnotations()
    Service-->>Caller: Annotation[]
```

```
Caller->>Service: toFragment()
Service-->>Caller: #atloria-annotations=...
Caller->>Viewer: Open screenshot URL with fragment
Viewer->>Viewer: Render overlay annotations
```

Usage

```
import { AnnotationEmitterService } from './annotation-emitter.service';

// Typically injected by NestJS with the current step/page context.
const annotationEmitter = new AnnotationEmitterService(/* dependencies */);

// Resolve the step target against the currently loaded live page.
const targetBox = await annotationEmitter.recomputeTargetBoxPct();

if (targetBox) {
  const annotations = annotationEmitter.buildStepAnnotations();
  const annotationFragment = annotationEmitter.toFragment();

  const viewerUrl =
    `https://viewer.example/screenshots/step-42.png${annotationFragment}`;

  console.log({ targetBox, annotations, viewerUrl });
}
```

AI Coding Instructions

- Keep annotations non-destructive: generate coordinate metadata and URL fragments only; never draw or alter screenshot pixels.
- Call `recomputeTargetBoxPct()` after the live page and target element are available, and handle `null` when an element cannot be found or measured.
- Preserve percentage-of-natural-image coordinates so annotations remain aligned when the viewer scales the screenshot.
- Use `buildStepAnnotations()` as the canonical overlay payload source, then pass `toFragment()` output directly to the viewer URL.
- Maintain compatibility with the existing Python producer behavior in `capture_annotations.py` and `emit_annotations.py` when changing coordinate or fragment serialization logic.

Referenced By

- `ScreenshotModule` (MODULE_PROVIDES)

- `FlowReplayService` (DEPENDS_ON)