SdkStorageService

September 4, 2026

SdkStorageService

Kind: Service

Source: `atloria-monorepo/apps/sdk-worker/src/app/sdk-gen/sdk-storage.service.ts`

Azure Blob upload for generated SDK zips (mirrors the screenshot-worker StorageService pattern: connection-string auth, auto-created public-read container, immutable cache headers — artifacts are content-addressed so a blob's content never changes under its name).

`SdkStorageService` uploads generated SDK ZIP artifacts to Azure Blob Storage. It uses connection-string authentication, creates the public-read container when needed, and applies immutable cache headers because SDK artifacts are content-addressed and never change for a given blob name.

Methods

Method	Signature	Returns	Description
`isAvailable`	`isAvailable()`	`boolean`	
`upload`	`upload(buffer: Buffer, blobName: string, contentType: unknown)`	`Promise<string>`	Upload a zip; returns its public (CDN when configured) URL.

Dependencies

- `ConfigService`

Where it refuses work

- `SdkStorageService` stops the work with `Error` when `!this.containerClient` — “Blob storage unavailable — cannot persist the SDK artifact”.

When something fails

- `SdkStorageService` handles failure in 1 place: it logs it and continues in all 1.

Diagram

```
sequenceDiagram
  participant Generator as SDK Generator
  participant Storage as SdkStorageService
  participant Azure as Azure Blob Storage

  Generator->>Storage: isAvailable()
  Storage-->>Generator: boolean

  Generator->>Storage: upload(zipBuffer, artifactName)
  Storage->>Azure: Create/get public-read container
  Azure-->>Storage: Container ready
  Storage->>Azure: Upload ZIP blob with immutable cache headers
  Azure-->>Storage: Blob URL
  Storage-->>Generator: Promise<string> artifact URL
```

Usage

```
import { SdkStorageService } from './sdk-gen/sdk-storage.service';

@Injectable()
export class SdkGenerationService {
  constructor(private readonly storage: SdkStorageService) {}

  async publishSdk(zipBuffer: Buffer, artifactName: string): Promise<string> {
    if (!this.storage.isAvailable()) {
      throw new Error('SDK artifact storage is not configured');
    }

    const downloadUrl = await this.storage.upload(zipBuffer, artifactName);

    return downloadUrl;
  }
}
```

AI Coding Instructions

- Check `isAvailable()` before attempting uploads so SDK generation can fail gracefully when Azure storage credentials are not configured.

- Keep blob names content-addressed (for example, based on an SDK version or content hash); do not overwrite an existing artifact under the same name.
- Preserve immutable cache headers on uploaded ZIP files, since consumers may cache artifact URLs indefinitely.
- Treat the URL returned by `upload()` as the public artifact download URL and persist or return it with SDK generation results.
- Keep Azure connection-string configuration aligned with the screenshot-worker storage pattern, including container creation and public-read access.

Relationships

- DEPENDS_ON → `configservice`

Referenced By

- `SdkGenModule` (MODULE_PROVIDES)