ChatGPT

Accessing ARK: Survival Ascended Game Data for a Breeding Manager

Building a breeding manager for **ARK: Survival Ascended (ASA)** requires retrieving detailed in-game data (creatures, tames, items, characters, etc.) through either files or live interfaces. Below we outline the available methods, from parsing save files in singleplayer to using mods or plugins for automated sync, along with relevant tools and the impact of ASA's Unreal Engine 5 upgrade on data access.

Official Data Access Methods in ASA

Out-of-the-box, ASA offers **no official REST API** for querying game state. The official "Web API" is extremely limited – it only serves game client needs like listing official servers and news, not in-game creature or player data [1]. There is a **remote console (RCON)** interface (as in ARK: Survival Evolved) which lets server admins run console commands remotely, but this is not a structured data API. RCON can fetch basic info (e.g. list online players or execute cheat commands) but cannot directly export all creature stats or inventories in a convenient format. In short, ASA inherits the same lack of a built-in data-export interface as its predecessor.

Reading Game Data from Save Files (Singleplayer & Server)

For singleplayer games or self-hosted servers, the **world save files** are a rich source of data. ASA uses binary save formats similar to ARK:SE: the main `.ark` world file captures the entire game state – all creatures (wild and tamed), player characters, inventories, items, structures, etc. – at the time of save [2]. In addition, each tribe and player have separate `.arktribe` and `.arkprofile` files storing tribe ownership info, player stats, engrams, and inventories. By parsing these files, one can extract **all relevant in-game data** needed for a breeding manager.

Community-developed save parsing tools have quickly emerged for ASA. For example, the open-source **ArkParse** library for Python can read and decode ASA save files, providing high-level APIs to access **detailed information about players, structures, equipment, dinosaurs, and more** [2]. ArkParse effectively exposes the game state in an object-oriented way – it lets you retrieve player and tribe data (including inventories), analyze structures by location or owner, inspect items/blueprints, and enumerate dinos with their stats (levels, mutations, etc.) [3]. In fact, ArkParse's features include a **Dino API** (to find specific tames or track their mutation counts and levels) and even a **JSON export** module for easily outputting the parsed data to structured JSON files [3] [4]. This makes it straightforward to feed the information into a web application.

Other tools mirror this approach: the **ARK Save Visualizer (ASV)**, originally made for ARK:SE, has been updated to support ASA saves [5]. ASV can load an `.ark` file and display all creatures (wild and tamed), their locations, stats, inventories, etc., and it provides an option to **export the data as JSON** for use in external apps [6]. This is useful for building web dashboards or integrating with a breeding manager. The **ark-sa-save-tools** project (Java) is another example, able to parse ASA world, tribe, and player files – it can already read most game object properties from the save [7]. In summary, **parsing ASA save files is a proven method** to get comprehensive data on creatures, items, and characters in a singleplayer or server context, with open-source libraries available to handle the binary format.

Tip: To use this in a Next.js web app, one approach is a user-initiated upload of their save files. With player consent, the save can be parsed server-side (e.g. via an API route or microservice using ArkParse or a similar library) to synchronize the breeding data. This avoids any need for direct game integration and works well for singleplayer or any scenario where the user can obtain the save files (for example, a self-hosted dedicated server).

In-Game Mods and Plugins for Live Data Sync

For a more automated or real-time sync – especially on a dedicated server – ASA's **modding system** or third-party server plugins can be leveraged. ASA introduced a new cross-platform modding ecosystem (distributed via CurseForge and an in-game mod manager, rather than the Steam Workshop) [8] [9]. While mods are typically used to add gameplay content, they *can* be designed to export game data with player permission.

Notably, ASA **does support external communication from mods**. A prime example is the **"Livemap (HTTPLocation)" mod**, which demonstrates that a mod can push game data to a web service. This server-side utility mod adds an item and, when active, **sends all players' locations to a specified web URL as an HTTP POST request with a JSON payload** [10]. Server owners configure the target endpoint in the INI settings, and the mod posts live positional data (the mod's CurseForge page even provides a JSON example and suggests using a service like Pipedream to inspect the incoming data) [11]. This confirms that **ASA's modding API allows HTTP requests** (likely via UE5 Blueprint or a plugin within the mod), enabling game data to be transmitted externally in real time. In the context of a breeding manager, one could develop a similar **server-side ASA mod** that collects breeding-related data – e.g. all tamed creatures and their stats on the server – and periodically sends this data (perhaps in batches as JSON) to your web application's API. Players/servers would install this mod and opt-in by configuring the target URL, thereby giving consent for their data to sync.

Another route on PC servers is the community-maintained **ARK Server API** (ASA edition). This is a **C++ plugin framework** that hooks into the server executable to extend its functionality. The ASA Server API has many plugins already (for admin tools, economy, etc.), and crucially it can facilitate external data exchange. For example, there's a donation plugin that links the ASA server to a web store – it listens for webhooks and injects in-game rewards via server commands, using secure HMAC-authenticated requests [12]. With this power, one could write a custom Server API plugin to **query all dinos/players in memory and expose them** (say, by writing to JSON files or opening a local web server or sending to an API endpoint). The Server API essentially gives full programmatic access to the server's data and the ability to run code (including network communication) in the server process. The downside is that this method is only viable for **self-hosted Windows servers** (it requires installing a DLL and is not officially supported by Wildcard). Many hosting providers (especially the official cross-play Nitrado servers) may not allow such plugins. Still, for private dedicated servers it's an option – and indeed ASA's Server API exists and supports dozens of plugins already [13] [14].

In summary, **automated data sync is feasible via mods or plugins**. A **mod** built with ASA's UE5 Dev Kit can run in-game and push consented data out (confirmed by community mods like Livemap that broadcast player stats via HTTP [10]). This approach works cross-platform if the mod is properly published (PC/console), although one must ensure performance is managed (sending thousands of creature records frequently could be heavy – batching or on-demand sync would be prudent). Meanwhile, **server plugins (Ark Server API)** offer even more control for PC servers, though at the cost of being unofficial and likely PC-only. Depending on your target deployment (singleplayer vs dedicated server vs hosted service), you might initially implement a **manual "sync" button** where users either

upload their save or run a provided script/mod to push data, and later consider a full live-sync mod once the ecosystem matures.

Community Tools and Resources

The ASA community has produced several tools that can help or inspire your project:

- **ARK Smart Breeding (ASB)** – This popular breeding tool from ARK:SE has been **updated to support ASA**. Players can use ASB in ASA mode to input or import creature stats and manage breeding plans. In fact, ASB can directly read dino stat exports from the game. In ARK:SE, one could export a creature's stats to a file via the in-game function (or use OCR/screenshot). In ASA, players report that **ASB works by simply switching the mode to ASA in the app's settings**, using updated stat formulas [15] [16]. There's even a **community mod "Ark Smart Breeding Export Gun"** on CurseForge that facilitates exporting creature stats in ASA, acting like a Tek Binocular to output stat data in ASB-compatible format [17]. While ASB itself is a standalone app (not a web app), it proves that all necessary breeding stats can be obtained from the game, and you could emulate similar extraction in your Next.js app via save parsing or a mod.

- **Save Parsing Libraries** – As noted, open-source libraries exist to parse ASA saves. Besides ArkParse (Python) and ark-sa-save-tools (Java) mentioned above, there are also tools in C# (the ArkSavegameToolkit used by ASV [18]) and others. These can be invaluable if you prefer to build a backend service that reads save files directly. For instance, ArkParse's GitHub README showcases how it can retrieve **nearly everything** from the save, including player inventories, structure locations, and dino genetics [2] [3], which covers all data needed for a breeding manager (creature stats, ancestry, mutations, etc.). Many of these projects also allow **writing or modifying saves**, though for your purposes read-only analysis is likely enough.

- **ASA Dev Kit Documentation** – If you pursue a mod solution, familiarize yourself with the ASA Dev Kit (Unreal Engine 5 based). Studio Wildcard has provided a streamlined Dev Kit for ASA with "**cloud cooking**" and cross-platform support [19] [20]. While specific documentation on data-access hooks is scarce, the modding community (forums, Discord) can offer guidance. For example, the existence of the Livemap mod suggests that UE5 **Blueprints may include an HTTP request capability or allow using an HTTP library** in mods. Investigating the Livemap mod's blueprint (if source is available) or asking mod authors could reveal how to implement a similar data export feature in your own mod.

- **Unreal Console Commands** – As an aside, ARK has various admin console commands (e.g. `GetAllState <ClassName>` which dumps all instances of a class to the log). These can technically be used via RCON or in-game to find creatures or data. However, parsing console log output is crude and not recommended for comprehensive data sync. It's mentioned here for completeness – a server admin *could* script something that uses RCON to periodically run commands and scrape results, but given the robust options above (save parsing, mods, APIs), this is usually unnecessary.

ASA and Unreal Engine 5: Impact on Data Access

ARK: Survival Ascended is built on **Unreal Engine 5**, a leap from the original's UE4, but this mostly affects graphics and performance rather than how data can be accessed. The switch to UE5 did not introduce any official data API; all **game state is still stored in memory and save files, not exposed via web services**. The data structures and formats are similar enough to ARK:SE's that community tools

were adapted rather quickly (with tweaks for UE5 serialization differences). One notable change is the **modding system overhaul**: mods are now delivered via a cross-platform system (no Steam Workshop) to support consoles [8]. This means that any **mod-based data collection** must abide by console compatibility – likely restricting the use of outright binary executables, but as demonstrated, **Blueprint-based mods can still perform HTTP requests** in ASA [10]. UE5's capabilities might even make certain things easier (for example, improved Blueprint interfaces or plugins), but the core approach (either read the **binary save** or run **in-game code** to emit data) remains the same as before.

In practical terms, **ASA's UE5 upgrade ensures that approaches proven in ARK:SE are still viable**, with updated tools. Save files (.ark, .arkprofile, .arktribe) continue to be the authoritative source for offline analysis [2]. Mods and server plugins continue to be the way to fetch live data in real-time, now enhanced by an official mod distribution that can reach console servers (with the caveat of platform limitations). The knowledge base from ARK:SE (RCON usage, admin commands, mod hooks) carries over, with community forums already discussing ASA-specific quirks and solutions.

References: The information above is drawn from ARK community wiki/documentation and proven community projects. ASA's official web API usage is documented to only cover server lists and news [1]. Save file contents and parsing are demonstrated by open-source projects like ArkParse (which accesses players, dinos, items from ASA saves) [2] [3] and posts on the ASA subreddit [7]. The ARK Save Visualizer tool confirms that one can export game data as JSON from saves for external use [6]. On the live data side, the Livemap mod's documentation shows ASA mods can perform HTTP POST with JSON payloads to external servers [10]. Additionally, the Ark Server API plugin listings illustrate external integrations (e.g. a donation webstore plugin) for ASA servers [12]. Finally, CurseForge's ASA modding announcements describe the new cross-platform modding experience on UE5 [9]. These sources collectively indicate that with some development effort, **all relevant ASA in-game data can be accessed** – either by reading the game's save data or by pushing data out via mods/plugins – enabling a Next.js-based breeding manager to function effectively.

---

[1] Web API - ARK: Survival Evolved Wiki
https://ark.fandom.com/wiki/Web_API

[2] [3] [4] GitHub - VincentHenauGithub/ark-save-parser: A python library to parse ark saves
https://github.com/VincentHenauGithub/ark-save-parser

[5] [18] GitHub - miragedmuk/ASV: ASV - Ark Save Visualisation and JSON export
https://github.com/miragedmuk/ASV

[6] ARK Savegame Visualiser (ASV) :: ARK: Survival Evolved General Discussions
https://steamcommunity.com/app/346110/discussions/0/3058490685009830068/

[7] Sharing ark-sa-save-tools, a parser for Ark: Survival Ascended save data : r/ARK
https://www.reddit.com/r/ARK/comments/17s1ys7/sharing_arksasavetools_a_parser_for_ark_survival/

[8] [9] [19] [20] ARK: Survival Ascended creators - Getting started: CurseForge support
https://support.curseforge.com/support/solutions/articles/9000232897-ark-survival-ascended-creators-getting-started

[10] [11] Livemap aka. HTTPLocation - Ark Survival Ascended Mods - CurseForge
https://www.curseforge.com/ark-survival-ascended/mods/livemap

[12] [13] [14] Ark Survival Ascended | Ark Server Api
https://ark-server-api.com/resources/categories/ark-survival-ascended.2/

15 16 ASA Smart Breeding ? : r/playark
https://www.reddit.com/r/playark/comments/1gcp635/asa_smart_breeding/

17 Ark Smart Breeding Export Gun - Ark Survival Ascended Mods
https://www.curseforge.com/ark-survival-ascended/mods/ark-smart-breeding-export-gun