# ArrayStore

DevExtreme’s **ArrayStore** is a client-side data store that manages JavaScript arrays efficiently. Below are key functionalities demonstrated:

#### **Initialization & Events**

* **data**: Stores dataset (marvelCharacters).
* **key**: Unique identifier (ID).
* Event handlers for loading, inserting, updating, deleting, modifying, and real-time push updates.

#### **Operations**

1. **Fetch Record** (byKey(key)) – Retrieves a record by ID.
2. **Insert Record** (insert(data)) – Adds a new record.
3. **Get Key Info** (key() & keyOf(object)) – Returns key properties.
4. **Load & Clear** (load() & clear()) – Loads data and clears the store.
5. **Push Changes** (push([{ type: "insert", data }])) – Adds records dynamically.
6. **Remove Record** (remove(key)) – Deletes a record by ID.
7. **Total Count** (totalCount()) – Retrieves the total number of records.
8. **Update Record** (update(key, data)) – Modifies an existing record.

# CustomStore

DevExtreme’s **CustomStore** enables CRUD operations with an external API. Key functionalities include:

#### **Initialization & Events**

* **key**: Unique identifier (id).
* **load**: Fetches data from API and displays it dynamically.
* **CRUD Events**: Triggers before/after inserting, updating, and deleting records.

#### **Operations**

1. **Fetch Data** (load()) – Loads data from API.
2. **Insert Record** (insert(values)) – Adds a new user.
3. **Update Record** (update(id, values)) – Modifies an existing user.
4. **Delete Record** (remove(id)) – Deletes a user.
5. **Real-Time Updates** (onPush(changes)) – Handles live data changes.

# DataSource

### Key Options:

* store – Data source (ArrayStore, CustomStore, etc.).
* filter – Filters data (e.g., ["ID", ">", 1]).
* sort – Sorts data ({ selector: "Name", desc: true }).
* group – Groups data by a field.
* paginate – Enables pagination (true/false).
* pageSize – Number of records per page.
* requireTotalCount – Fetches the total count of records.
* searchExpr – Field(s) to search within.
* searchOperation – Defines search behavior (contains, startswith).

### Methods:

* load() – Fetches data.
* reload() – Reloads data.
* filter(expr) – Gets/Sets filter.
* sort(expr) – Gets/Sets sorting.
* group(expr) – Gets/Sets grouping.
* pageIndex(value) – Gets/Sets page index.
* pageSize(value) – Gets/Sets page size.
* totalCount() – Returns the total record count.
* isLoading() – Checks if data is loading.

### Key Events:

* changed – Fires when data is changed.
* loadError – Fires if data loading fails.
* loadingChanged – Fires when the loading state changes.

# LocalStore

### Key Options:

* data – Stores local data.
* key – Unique identifier for records.
* name – Local storage key name.
* immediate – Saves data immediately (true/false).

### Methods:

* load() – Loads data from LocalStore.
* insert(values) – Adds a new record.
* update(key, values) – Updates an existing record.
* remove(key) – Removes a record by key.
* clear() – Clears all stored data.
* totalCount() – Returns the total number of records.

### Key Events:

* loaded – Fires when data is loaded.
* inserting / inserted – Fires before/after inserting a record.
* updating / updated – Fires before/after updating a record.
* removing / removed – Fires before/after removing a record.
* modified – Fires when data is changed.

# Query

### Key Methods:

* filter(criteria) – Filters data using a condition.
* sortBy(getter, desc?) – Sorts data (optional descending order).
* groupBy(getter) – Groups data by a specific field.
* select(getter) – Selects specific fields from data.
* slice(skip, take) – Skips & limits records (pagination).
* count() – Returns the total number of records.
* sum(getter?) – Calculates the sum of a field.
* avg(getter?) – Returns the average value of a field.
* min(getter?) / max(getter?) – Finds the min/max value.
* toArray() – Converts the query result into an array.

### Advanced Methods:

* aggregate(step, finalize?) – Performs custom aggregation.
* enumerate() – Executes the query and returns results.
* thenBy(getter, desc?) – Sorts by a secondary field.