## Data Layer

### Array Store

The ArrayStore is a store that provides an interface for loading and editing an in-memory array and handling related events.

#### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property/Function** | **Description** | **Default Value** | **Available Values** | **Input Type** |
| **data** | Specifies the store's associated array. | [] (empty array) | Array of objects | Array |
| **errorHandler** | Function executed when the store throws an error. | null | Any valid function | Function |
| **key** | Unique key property/properties to access data items. | null | Any string or array of strings representing object keys | String / Array |
| **onInserted** | Function executed after a data item is added. | null | Any valid function | Function |
| **onInserting** | Function executed before a data item is added. | null | Any valid function | Function |
| **onLoaded** | Function executed after data is loaded. | null | Any valid function | Function |
| **onLoading** | Function executed before data is loaded. | null | Any valid function | Function |
| **onModified** | Function executed after an item is added, updated, or removed. | null | Any valid function | Function |
| **onModifying** | Function executed before an item is added, updated, or removed. | null | Any valid function | Function |
| **onPush** | Function executed before changes are pushed to the store. | null | Any valid function | Function |
| **onRemoved** | Function executed after a data item is removed. | null | Any valid function | Function |
| **onRemoving** | Function executed before a data item is removed. | null | Any valid function | Function |
| **onUpdated** | Function executed after a data item is updated. | null | Any valid function | Function |
| **onUpdating** | Function executed before a data item is updated. | null | Any valid function | Function |

* byKey():
  + Fetch a specific record using a unique key or a combination of keys.
  + Look up records efficiently in an **ArrayStore**.
  + Read a single item before updating or deleting CRUD operations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Parameters** | **Return Type** |
| **byKey(key)** | Retrieves a data item with a specific key. | key: The key value of the item. | Promise<object> |
| **clear()** | Clear all the ArrayStore's associated data. | None | void |
| **createQuery()** | Creates a Query object to filter, sort, group, or process data in the array. | None | Query |
| **insert(values)** | Add a data item to the store. | values: The data object to insert. | Promise<object> (Inserted item) |
| **key()** | Returns the name(s) of the key field(s) used to identify unique records in the store. | None | string |
| **keyOf(obj)** | Extracts the key value from a given data object. | obj: The data object. | string |
| **load()** | Fetches all data as stored in the ArrayStore. | None | Promise<object[]> (Array of data items) |
| **load(options)** | Fetches data based on custom conditions like filtering, sorting, paging, etc. | options: Load options like filter, sort, take, skip, etc. | Promise<object[]> |
| **push(changes)** | Pushes data changes to the store and notifies the DataSource. | changes: Array of changes { type, key, data }. | void |
| **remove(key)** | Removes a data item with a specific key. | key: The key value of the item to remove. | Promise<void> |
| **totalCount(options)** | Gets the total count of items the load() function returns. | options: Load options like filters. | Promise<number> |
| **update(key, values)** | Updates a data item with a specific key. | key: The key of the item to update. values: The new values for the item. | Promise<object> (Updated item) |

### Custom Store

The CustomStore enables you to implement custom data access logic for consuming data from any source.

#### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property/Method** | **Description** | **Default Value** | **Available Values** | **Input Type** |
| **byKey** | Retrieves a data item with a specific key. | null | Function returning a Promise<object> | Function |
| **cacheRawData** | Specifies whether raw data should be saved in the cache. (Applies only if loadMode is "raw".) | false | true, false | Boolean |
| **errorHandler** | Function executed when the store throws an error. | null | Any valid function | Function |
| **insert** | Specifies a custom implementation of insert(values). | null | Function returning a Promise<object> | Function |
| **key** | Unique key property (or properties) to access data items. | null | Any string or array of strings representing object keys | String / Array |
| **load** | Specifies a custom implementation of load(options). | null | Function returning a Promise<object[]> | Function |
| **loadMode** | Specifies how data returned by the load function is treated. | "processed" | "processed", "raw" | String |
| **remove** | Specifies a custom implementation of remove(key). | null | Function returning a Promise<void> | Function |
| **totalCount** | Specifies a custom implementation of totalCount(options). | null | Function returning a Promise<number> | Function |
| **update** | Specifies a custom implementation of update(key, values). | null | Function returning a Promise<object> | Function |
| **useDefaultSearch** | Specifies whether the store combines search and filter expressions. Defaults to true if loadMode is "raw" and false if "processed". | Depends on loadMode | true, false | Boolean |

#### Methods

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Parameters** | **Return Type** |
| **clearRawDataCache()** | Deletes data from the cache. Takes effect only if cacheRawData is true. | None | void |

#### Load Options

This object is used to specify settings according to which the server should process data. More often these settings are passed as a parameter to the load function and depend on the operations (paging, filtering, sorting, etc.) that you have enabled in the DataSource or UI component.

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Available Values** | **Input Type** |
| **filter** | A filter expression used to filter data. | Array, function, or filter expression | Array / Function / Expression |
| **group** | A group expression. Defines grouping rules. | { selector, desc, isExpanded, groupInterval } | Object |
| **groupSummary** | A summary expression used with the group setting. | { selector: "field", summaryType: "sum" } | Object |
| **parentIds** | The IDs of the rows being expanded. Used in the **TreeList** UI component. | Array of IDs | Array |
| **requireGroupCount** | Indicates whether a top-level group count is required. Works with filter, take, skip, requireTotalCount, and group settings. | true, false | Boolean |
| **requireTotalCount** | Indicates whether the total count of data objects is needed. | true, false | Boolean |
| **searchExpr** | A data field or expression whose value is compared to the search value. | Field name or expression | String / Function / Array |
| **searchOperation** | The comparison operation for searching. | "=", "<> ", ">", ">=", "<", "<=", "startswith", "endswith", "contains", "notcontains", "isblank", "isnotblank" | String |
| **searchValue** | The current search value for filtering. | Any data value | String / Number / Boolean |
| **select** | A select expression that determines which fields to load. | Array of field names | Array |
| **skip** | The number of records to skip from the result set. Used for paging. | Positive integer | Number |
| **sort** | A sort expression that defines sorting rules. | { selector, desc } | Object / Array |
| **take** | The number of data objects to load. Used for paging with skip. | Positive integer | Number |
| **totalSummary** | A summary expression for total calculations. | Array of summary expressions | Array |
| **userData** | An object for storing additional settings to be sent to the server. | Any key-value object | Object |

### Data Source

The DataSource is an object that provides an API for processing data from an underlying store.

#### Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Description** | **Default Value** | **Available Values** | **Input Type** |
| **map** | Specifies an item mapping function. | null | Function | Function |
| **onChanged** | Function executed after data is loaded. | null | Function | Function |
| **onLoadError** | Function executed when data loading fails. | null | Function | Function |
| **onLoadingChanged** | Function executed when the data loading status changes. | null | Function | Function |
| **pageSize** | Maximum number of data items per page (applies only if paginate is true). | null | Positive integer | Number |
| **paginate** | Specifies whether the DataSource loads data in pages or all at once. | true (if group is not set), false (if group is set) | true, false | Boolean |
| **postProcess** | Specifies a post-processing function. | null | Function | Function |
| **pushAggregationTimeout** | Time (in ms) before aggregating changes and pushing them to the DataSource. | null | Positive integer | Number |
| **reshapeOnPush** | Reapplies sorting, filtering, and grouping after receiving a push. | false | true, false | Boolean |
| **store** | Configures the underlying store of the DataSource. | null | `{ type: "array" | "local" |

#### Methods

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Parameters** | **Return Type** |
| **cancel(operationId)** | Cancels the load operation with a specific identifier. | operationId (String/Number) | void |
| **filter()** | Gets the filter property's value. | None | Array / Function |
| **filter(filterExpr)** | Sets the filter property's value. | filterExpr (Array/Function) | void |
| **group()** | Gets the group property's value. | None | Array / Object |
| **group(groupExpr)** | Sets the group property's value. | groupExpr (Array/Object) | void |
| **isLastPage()** | Checks if the number of items on the current page is less than pageSize. Works only when paging is enabled. | None | Boolean |
| **isLoaded()** | Checks whether data is loaded in the DataSource. | None | Boolean |
| **isLoading()** | Checks whether data is being loaded in the DataSource. | None | Boolean |
| **items()** | Gets an array of data items on the current page. | None | Array |
| **key()** | Gets the key property value of the underlying store. | None | String / Array |
| **load()** | Starts loading data. | None | Promise |
| **loadOptions()** | Gets an object with current data processing settings. | None | Object |
| **pageIndex()** | Gets the current page index. | None | Number |
| **pageIndex(newIndex)** | Sets the index of the page that should be loaded in the next load() call. | newIndex (Number) | void |
| **pageSize()** | Gets the page size. | None | Number |
| **pageSize(value)** | Sets the page size. | value (Number) | void |
| **paginate()** | Gets the paginate property's value. | None | Boolean |
| **paginate(value)** | Sets the paginate property's value. | value (Boolean) | void |
| **reload()** | Clears currently loaded DataSource items and calls load(). | None | void |
| **requireTotalCount()** | Gets the requireTotalCount property's value. | None | Boolean |
| **requireTotalCount(value)** | Sets the requireTotalCount property's value. | value (Boolean) | void |
| **searchExpr()** | Gets the searchExpr property's value. | None | String / Array |
| **searchExpr(expr)** | Sets the searchExpr property's value. | expr (String/Array) | void |
| **searchOperation()** | Gets the searchOperation property's value. | None | String |
| **searchOperation(op)** | Sets the searchOperation property's value. | op (String) | void |
| **searchValue()** | Gets the searchValue property's value. | None | Any |
| **searchValue(value)** | Sets the searchValue property's value. | value (Any) | void |
| **select()** | Gets the select property's value. | None | Array |
| **select(expr)** | Sets the select property's value. | expr (Array) | void |
| **sort()** | Gets the sort property's value. | None | Array |
| **sort(sortExpr)** | Sets the sort property's value. | sortExpr (Array) | void |
| **store()** | Gets the instance of the store underlying the DataSource. | None | Object |
| **totalCount()** | Gets the number of data items in the store after the last load() operation without paging. Works only if requireTotalCount is true. | None | Number |

### Local Store

#### Properties

The LocalStore is a store that provides an interface for loading and editing data from HTML Web Storage (also known as window.localStorage) and handling related events.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Description** | **Default Value** | **Other Available Values** | **Input Type** |
| **data** | Specifies the store's associated array. | [] | Any array | Array |
| **errorHandler** | Specifies the function that is executed when the store throws an error. | null | Function reference | Function |
| **flushInterval** | Specifies a delay (in ms) before changes are saved in local storage. | 1000 | Any positive number | Number |
| **immediate** | Determines if LocalStore saves changes immediately. | true | false | Boolean |
| **key** | Specifies the key property (or properties) for data access. Each key value must be unique. | "" | Any string or array of strings | String/Array |
| **name** | Specifies the local storage key name (prefixed with dx-data-localStore-). | "" | Any string | String |
| **onInserted** | A function executed after a data item is added to the store. | null | Function reference | Function |
| **onInserting** | A function executed before a data item is added to the store. | null | Function reference | Function |
| **onLoaded** | A function executed after data is loaded to the store. | null | Function reference | Function |
| **onLoading** | A function executed before data is loaded to the store. | null | Function reference | Function |
| **onModifying** | A function executed before a data item is added, updated, or removed from the store. | null | Function reference | Function |
| **onModified** | A function executed after a data item is added, updated, or removed from the store. | null | Function reference | Function |
| **onPush** | A function executed before changes are pushed to the store. | null | Function reference | Function |
| **onRemoved** | A function executed after a data item is removed from the store. | null | Function reference | Function |
| **onRemoving** | A function executed before a data item is removed from the store. | null | Function reference | Function |
| **onUpdated** | A function executed after a data item is updated in the store. | null | Function reference | Function |
| **onUpdating** | A function executed before a data item is updated in the store. | null | Function reference | Function |

#### Methods

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Parameters** | **Return Type** |
| **byKey(key)** | Retrieves a data item with the specified key. | key: any | Object |
| **clear()** | Removes all data from the local storage. | None | void |
| **createQuery()** | Creates a Query for the underlying array. | None | Query |
| **insert(values)** | Adds a data item to the store. | values: Object | Promise<Object> |
| **key()** | Retrieves the key property (or properties) specified for the store. | None | String/Array |
| **keyOf(obj)** | Gets the key value of a given data item. | obj: Object | any |
| **load()** | Loads data from the store. | None | Promise<Array> |
| **load(options)** | Loads data with specific options. | options: Object | Promise<Array> |
| **push(changes)** | Pushes data changes to the store. | changes: Array | void |
| **remove(key)** | Removes a data item with a specific key. | key: any | Promise<void> |
| **totalCount(options)** | Retrieves the total number of data items. | options: Object | Promise<number> |
| **update(key, values)** | Updates a data item with a specific key. | key: any, values: Object | Promise<Object> |

### Query

The Query is an object that provides a chainable interface for making data queries.

To create a Query, call the query(array) or query(url, queryOptions) method, depending on the type of the storage you access. The Query supports method chaining. This enables you to execute several methods in a single statement.

#### Methods

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Parameters** | **Return Type** |
| **aggregate(seed, step, finalize)** | Calculates a custom summary for all data items. | seed: any, step: Function, finalize: Function | any |
| **aggregate(step)** | Calculates a custom summary for all data items. | step: Function | any |
| **avg()** | Calculates the average of all values. Applies only to numeric arrays. | None | number |
| **avg(getter)** | Calculates the average of all values found using a getter. | getter: Function | number |
| **count()** | Calculates the number of data items. | None | number |
| **enumerate()** | Executes the Query asynchronously. | None | Promise<Array> |
| **filter(criteria)** | Filters data items using a filter expression. | criteria: any | Query |
| **filter(predicate)** | Filters data items using a custom function. | predicate: Function | Query |
| **groupBy(getter)** | Groups data items by the specified getter. | getter: Function | Query |
| **max()** | Calculates the maximum value. Applies only to numeric arrays. | None | number |
| **max(getter)** | Calculates the maximum of all values found using a getter. | getter: Function | number |
| **min()** | Calculates the minimum value. Applies only to numeric arrays. | None | number |
| **min(getter)** | Calculates the minimum of all values found using a getter. | getter: Function | number |
| **select(getter)** | Selects individual fields from data objects. | getter: Function | Query |
| **slice(skip, take)** | Gets a specified number of data items starting from a given index. | skip: number, take: number | Query |
| **sortBy(getter)** | Sorts data items by the specified getter in ascending order. | getter: Function | Query |
| **sortBy(getter, desc)** | Sorts data items by the specified getter in the specified order. | getter: Function, desc: boolean | Query |
| **sum()** | Calculates the sum of all values. | None | number |
| **sum(getter)** | Calculates the sum of all values found using a getter. | getter: Function | number |
| **thenBy(getter)** | Sorts data items by an additional getter in ascending order. | getter: Function | Query |
| **thenBy(getter, desc)** | Sorts data items by an additional getter in the specified order. | getter: Function, desc: boolean | Query |
| **toArray()** | Gets data items associated with the Query synchronously. | None | Array |