



\$app/forms

ON THIS PAGE



```
import { applyAction, deserialize, enhance } from '$app/forms';
```

applyAction

This action updates the `form` property of the current page with the given data and updates `$page.status`. In case of an error, it redirects to the nearest error page.

```
function applyAction<
  Success extends Record<string, unknown> | undefined,
  Failure extends Record<string, unknown> | undefined
>(
  result: import('@sveltejs/kit').ActionResult<
    Success,
    Failure
  >
): Promise<void>;
```

deserialize

Use this function to deserialize the response from a form submission. Usage:

```
import { deserialize } from '$app/forms';

async function handleSubmit(event) {
  const response = await fetch('/form?action', {
```



```
const result = deserialize(await response.text());
// ...
}
```

```
function deserialize<
  Success extends Record<string, unknown> | undefined,
  Failure extends Record<string, unknown> | undefined
>(
  result: string
): import('@sveltejs/kit').ActionResult<Success, Failure>;
```

enhance

This action enhances a `<form>` element that otherwise would work without JavaScript.

The `submit` function is called upon submission with the given `FormData` and the `action` that should be triggered. If `cancel` is called, the form will not be submitted. You can use the `abort controller` to cancel the submission in case another one starts. If a function is returned, that function is called with the response from the server. If nothing is returned, the fallback will be used.

If this function or its return value isn't set, it

- falls back to updating the `form` prop with the returned data if the action is on the same page as the form

- updates `$page.status`

- resets the `<form>` element and invalidates all data in case of successful submission with no redirect response

- redirects in case of a redirect response

- redirects to the nearest error page in case of an unexpected error

If you provide a custom function with a callback and want to use the default behavior,

```
Success extends Record<string, unknown> | undefined,  
Failure extends Record<string, unknown> | undefined  
>(  
  form_element: HTMLFormElement,  
  submit?: import('@sveltejs/kit').SubmitFunction<  
    Success,  
    Failure  
  >  
) : {  
  destroy(): void;  
};
```

[✎ Edit this page on GitHub](#)

PREVIOUS

[\\$app/environment](#)

NEXT

[\\$app/navigation](#)