

# 【Larvel】 redirect() 與 redirect()->intended()

在 Laravel 中，`redirect()->intended('/login')` 和 `redirect('/login')` 有一些差異，主要是用於不同的用途和情境：

1. **`redirect()->intended('/login')`：**

- `intended()` 方法用於重定向到上一個受保護路由的預期 URL。這通常在使用中間件保護的路由中，當用戶未登入而被重定向到登入頁面時，用於指示登入成功後應該返回的路由。
- 當用戶在未登入狀態下訪問一個需要登入才能訪問的路由時，系統會記錄下該受保護路由的 URL。當用戶成功登入後，使用 `redirect()->intended('/login')` 將用戶重定向到該受保護路由的 URL，這樣可以提供更好的使用者體驗。

```
return redirect()->intended('/login');
```

2. **`redirect('/login')`：**

- `redirect('/login')` 簡單地將用戶重定向到指定的 URL，這個 URL 可以是任何您希望將用戶重定向到的地方，例如登入頁面或任何其他頁面。

```
return redirect('/login');
```

## 差異及適用情境：

- **使用 `redirect()->intended('/login')`：**
  - 主要用於處理登入成功後的重定向，確保用戶可以返回他們原本想要訪問但需要登入才能進入的頁面。
  - 通常與中間件（如 `auth` 中間件）結合使用，當用戶在未登入狀態下訪問受保護路由時會被重定向到登入頁面，成功登入後，會使用 `intended()` 方法將用戶重定向回原本要訪問的受保護路由。
- **使用 `redirect('/login')`：**
  - 簡單地將用戶重定向到指定的 URL，通常用於靜態的重定向，不涉及登入成功後的動態路由記錄和重定向。

## 總結：

使用 `redirect()->intended('/login')` 和 `redirect('/login')` 主要取決於您的需求。如果您需要確保用戶在成功登入後返回到原本要訪問的受保護路由，應使用 `intended()` 方法。而如果僅僅需要將用戶重定向到靜態的登入頁面或其他固定的 URL，則可以直接使用 `redirect('/login')`。

---

🕒 修訂版本 #1

★ 由 treeman 建立於 6 📅 2024 16:20:17

✍ 由 treeman 更新於 6 📅 2024 16:22:19