

STL Model Download Automation for Dental Clinic Automation Project Using Python, Selenium, and PyAutoGUI

1. CONTEXT AND PROBLEM

Challenge:

- Time-consuming manual process for downloading and renaming STL models (±30 minutes daily)
- Frequent errors in naming patient folders
- Delays in making files available to dentists
- Operational overload due to repetitive tasks
- Time wasted with calls from dentists requesting unavailable files

2. DEVELOPED SOLUTION

Technologies Used:

- Python as the base programming language
- Selenium for web browser automation
- PyAutoGUI for desktop interface interaction
- Windows Task Scheduler for automated execution

Implemented Features:

- Automatic login to the scanner system
- Sequential download of STL models
- Automatic renaming of folders with correct patient information
- Scheduled daily execution at 8:00 PM after work hours
- Operation without the need for human intervention

3. TECHNICAL CHALLENGES AND SOLUTIONS

Challenges:

- Interaction with two different environments (web and desktop)
- Complexity of the initial monolithic code
- Precise mapping of the required sequence of actions

Solutions:

- Refactoring using the Page Object Model (POM) pattern
- Code modularization for better maintainability
- Detailed mapping of each step in the process
- Implementation of exception handling for greater robustness

4. RESULTS AND IMPACT

Quantitative Benefits:

- Saves 30 minutes of manual work daily (± 10 hours per month)
- 100% elimination of file naming errors
- STL models made available the same day as scanning

Qualitative Benefits:

- Staff redirected to higher-value tasks
- Significant reduction in calls from dentists requesting files
- Improved satisfaction among partner dentists
- Optimization of the clinic's workflow
- Reduced stress on the support team

5. LESSONS LEARNED

- Importance of detailed process mapping before automation
- Value of refactoring for code maintainability
- Benefits of modularization using the POM pattern
- Positive impact of automation beyond time savings
- Efficient integration between web and desktop automation tools

PT/BR

Automação de Download de Modelos STL para Clínica Odontológica

Projeto de Automação com Python, Selenium e PyAutoGUI

1. CONTEXTO E PROBLEMA

Desafio:

- Processo manual demorado de download e renomeação de modelos STL (± 30 min diários)
- Erros frequentes na nomeação das pastas de pacientes
- Atrasos na disponibilização dos arquivos para os dentistas
- Sobrecarga operacional da equipe com tarefas repetitivas
- Tempo perdido com ligações de dentistas solicitando arquivos indisponíveis

2. SOLUÇÃO DESENVOLVIDA

Tecnologias utilizadas:

- Python como linguagem base
- Selenium para automação de navegador web
- PyAutoGUI para interação com interfaces de desktop
- Agendador de tarefas do Windows para execução automatizada

Funcionalidades implementadas:

- Login automático no sistema do scanner
- Download sequencial dos modelos STL
- Renomeação automática das pastas com informações corretas dos pacientes
- Execução programada diária às 20h após o expediente
- Operação sem necessidade de intervenção humana

3. DESAFIOS TÉCNICOS E SOLUÇÕES

Desafios:

- Interação com dois ambientes distintos (web e desktop)
- Complexidade do código inicial monolítico
- Mapeamento preciso da sequência de ações necessárias

Soluções:

- Refatoração utilizando o padrão Page Object Model (POM)
- Modularização do código para melhor manutenção

- Mapeamento detalhado de cada etapa do processo
- Implementação de tratamentos de exceção para maior robustez

4. RESULTADOS E IMPACTO

Benefícios quantitativos:

- Economia de 30 minutos diários de trabalho manual (± 10 horas mensais)
- Eliminação de 100% dos erros de nomeação de arquivos
- Disponibilização dos modelos no mesmo dia do escaneamento

Benefícios qualitativos:

- Redirecionamento do funcionário para tarefas de maior valor
- Diminuição significativa de ligações de dentistas solicitando arquivos
- Melhoria na satisfação dos dentistas parceiros
- Otimização do fluxo de trabalho da clínica
- Redução do estresse da equipe de atendimento

5. LIÇÕES APRENDIDAS

- Importância do mapeamento detalhado de processos antes da automação
- Valor da refatoração para manutenibilidade do código
- Benefícios da modularização através do padrão POM
- Impacto positivo da automação além da economia de tempo
- Integração eficiente entre ferramentas de automação web e desktop